IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2019

From purchase, usage, to upgrade — Consumer analytics using large scale transactional data

Xinxue Qu
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

 Part of the Databases and Information Systems Commons

Recommended Citation
Qu, Xinxue, "From purchase, usage, to upgrade — Consumer analytics using large scale transactional data" (2019). *Graduate Theses and Dissertations*. 17079.
https://lib.dr.iastate.edu/etd/17079

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

From purchase, usage, to upgrade
— Consumer analytics using large scale transactional data

by

Xinxue Qu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Business and Technology (Information Systems)

Program of Study Committee:
Zhengrui Jiang, Major Professor
Joey F George
Abhay N Mishra
Zhu Zhang
Wei Zhang

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

Copyright © Xinxue Qu, 2019. All rights reserved.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Zhengrui Jiang, and my committee members, Dr. Joey F George, Dr. Abhay N Mishra, Dr. Zhu Zhang, and Dr. Wei Zhang, for their guidance and support throughout my five years' study at the Ivy College of Business, Iowa State University.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

ABSTRACT

The amount of data businesses are collecting about their customers is staggering. Firms can now easily track and record past purchases, product usage patterns, and customers' responses to marketing campaigns and promotion programs. If fully analyzed, such rich transactional data offers companies the opportunity to understand what drives customers' purchase decisions, how to improve consumers' shopping experience, and how to develop and retain loyal customers. My dissertation addresses these issues by applying consumer analytics, including association rule mining, survival analysis, econometrics, and optimization, on large-scale transactional data to help companies better understand, predict, and subsequently influence the consumption behavior of their customers.

My dissertation comprises three essays. The first essay utilizes multi-level association rule mining and proposes a project-oriented recommendation method to predict next purchases for inexperienced consumers. In the second essay, I propose an Expo-Decay proportional hazard model and use customers' adoptions and usage of previous product generations to understand and predict their upgrade behaviors for the current product generation. In the third essay, a time-based dynamic synchronization policy is applied for the maintenance of consolidated data repository under an infinite planning horizon.

In these essays, I apply and extend a variety of business analytics tools including data mining (association rule mining and collaborative filtering), survival analysis, dynamic programming, simulation, and econometric methods. These essays contribute to the consumer analytics literature and can help firms maintain high-quality data assets and

make informed decisions on cross-generation product development, product promotion and recommendation, and customer retention.

CHAPTER 1. GENERAL INTRODUCTION

Analytics has become a new source of competitive advantage for organizations. For the past few decades, organizations have implemented various information technologies to support business operations and decision makings. In the age of *Big Data*, the amount of information businesses have collected is increasing exponentially. The development of advanced analytic tools provides organizations more opportunities to utilize the information resources to gain more competitive advantages. However, consumers' decision making processes (including first time purchases and product/service upgrades) are not fully discovered. My dissertation extends our understanding about consumers' decision-makings and develops consumer analytic tools to better understand, predict, and subsequently influence the consumption behavior of consumers.

My dissertation comprises three essays. The first essay develops a recommendation method to predict project-oriented purchases to assist inexperienced consumers to make their decisions. The recommendation is based on frequently purchased product sets learnt from large-scale transactional records. The second essay proposes a survival analysis framework to understand and predict existing users' product upgrade decisions and identify experience-related factors that impact the upgrade behavior of existing users. To achieve a high accuracy of prediction and precisely understand consumers' decision makings, the quality of data collected significantly influences the quality of results generated by these analytic tools. Therefore, the third essay extends a time-based dynamic synchronization policy to an infinite planning horizon to schedule the maintenance of consolidated data repository by striking a balance between synchronization costs and losses incurred by poor decisions made on the low quality information.

A variety of business analytics tools have been applied and extended in my dissertation. In the first essay, association rule mining algorithms are applied to search for frequently purchase products from the historical transaction records, and then the collaborative filtering method is adopted to utilize the product category hierarchical information in the recommendation model. The evaluation is based on Top-N item prediction using precision and recall. In the second essay, based on the survival analysis (specifically the proportional hazard mode), I propose an Exponential-Decay proportional hazard model to estimate and predict the upgrade timing of existing users. Random effect model and point process model extensions are developed based on the baseline framework to address some econometric concerns. In the third essay, the consolidated data repository maintenance is modeled as a Markov decision process, and the dynamic programming method has been applied to develop an algorithm to search the optimal control limits. In policy comparisons, simulations are conducted to evaluate the cost-savings with benchmark policies.

My dissertation contributes to the consumer analytics literature by proposing various analytics tools to understand and predict consumers decision makings, from first-time purchase to product upgrade decisions. Moreover, an optimal data repository maintenance policy developed to address the data quality issues in analytics applications. These findings also carry some managerial implications. The recommendation method shows the importance of product category information and also provides a solution to data sparsity issues when dealing with transactional data in practice. The upgrade timing analysis can help companies with customer retention. The importance of previous adoption and usage experience provides organizations new insights on customer segmentation and target marketing. The consolidated

data repository maintenance policy provides data-driven analysis or research with some implications about the importance of data quality in decision makings.

In summary, my dissertation applies consumer analytics, including data mining, statistical learning, econometrics and optimization, on large-scale transaction records to help companies better understand consumers' decision makings, and subsequently predict and influence their consumption behavior.

CHAPTER 2. PROJECT-ORIENTED RECOMMENDATION BASED ON ASSOCIATION RULES

Modified from a manuscript to be submitted to INFORMS Journal on Computing

Xinxue Qu, Zhengrui Jiang, and Zhu Zhang

Ivy College of Business, Iowa State University, Ames, IA, USA

Abstract

The recommender system has been used as a tool in E-commerce for a long time. There are a lot popular algorithms generating reliable recommendations, like *collaborative filtering* and *association-rule-based method*. However, "*Buying mistakes*" are still the top 1 retail pain. The assumption that a customer's future purchase can be predicted using his/her taste/preference retrieved from the historic record may not hold when the customer's purchase is driven by the on-going project. The products purchased previously may have very low similarity with products needed in the future, which means the traditional collaborative filtering method will not work well in the project-driven scenario. Moreover, in real world, when the company has a great variety of products, the data sparsity problem makes association-rule-based method less useful. Therefore, in this study, we propose an association-rule-based method by utilizing the product category information to measure the similarity between products sets and generate high-quality project-oriented recommendations for customers.

Keywords: Recommender System, Association Rule Mining, Collaborative Filtering

Introduction

According to the latest report from Gartner, Inc., the global revenue in the business intelligence (BI) and analytics market is forecast to reach $16.9 billion in 2016, an increase of 5.2 percent from 2015. The great potential in the analytics market is inspired by the abundant data related to the customers or agents in business. For decades, companies have accumulated huge amount of information related to the customers and the business process. Recently, with the data mining and machine learning techniques, the business analytics toolkits can help squeeze information out from each bit of data stored, which can be directly linked to marketing profitability and efficiency (Blattberg et al. 2008).

The recommendation system is one of the widely applied toolkit for firms customize their marketing effort to targeted consumers. Matching the customers with the most appropriate products will enhance consumer satisfaction and loyalty. According to Hosanagar et al. (2014), 60% of the Netflix rentals stem from recommendations, and 35% of Amazon's sales are generated by their recommendation system. Among the recommendation algorithms, the most popular ones are content-based filtering, collaborative filtering (e.g. matrix factorization), and association-rule-based methods (Ghoshal et al. 2015). In this study, we mainly focus on providing high quality association-rule-based recommendations.

However, with the recommendations from these popular systems, customers still make the "wrong" purchasing decisions in the shopping trips. Every year, product returns cost U.S. manufacturers and retailers $100 billion in lost sales, transportation, handling, processing and disposal. Customer returns can reduce a manufacturer's profitability by an average of 3.8%. (Blanchard D. 2007) According to the recent study by Petersen and Kumar (2015), the percentage of customers who returned a product during their relationship with the firm at the time of the survey was relatively high. For a catalog apparel retailer, 70% of

customers returned a product; for a high-tech business-to-business firm, 64% of customers returned a product; and for a general merchandise retailer, 75% of customers returned a product. Even professional customers make mistakes in purchasing products for a project. (In this study, we find 99.6% of professional consumers have returned, while 48% regular consumers have returned.) Therefore, more work need to be done to improve the quality of recommendations.

Market basket analysis is a well-known data mining technique by studying what products consumers purchase together, which could help retailers discover cross and up-selling opportunities, develop promotions, determine product placement, and optimize the inventory. (Askar 2016) Using data mining techniques to analyze shopping basket data can help better understand customers' purchase behaviors and empower the customers to buy "smarter". According to the interviews by the Fact Point, among the over 50 retailers with revenues from $400 million to $24 billion, "Buying mistakes" are the top 1 retail pain (Gordon 2008). The primary objective of this study is to use the information from the potential ongoing projects hidden in the transaction records, and predict the future project-driven purchases.

To help the consumers identify what they really need, this study will develop a recommendation method using the on-going project information based on the association rules. Since each shopping basket is composed of products that are oriented toward one or multiple potential ongoing "project", the recommendation will rely on the similarity computed using the product(set)-based collaborative filtering methods. Meanwhile, the product category structure will be utilized to help measure the similarity between products sets and existing shopping baskets.

The related literature will be discussed next. Then we will describe the problem to be solved in detail. After that, we will propose the project-oriented recommendation method. Then, we will show the description of the data and the experiment results. We will conclude this paper with discussions

Related Studies

There are a variety of recommendation systems used in the E-commerce platforms by utilizing the demographic information, the content, or the historical records etc. The most successful and widely accepted methods are collaborative filtering (with matrix factorization techniques) and association rule mining.

Collaborative Filtering

Collaborative Filtering (CF) is firstly used in an email filtering system, Tapestry, by Goldberg et al. (1992). Different from the content filtering, which creates profiles for users and products to characterize the nature and uses the profiles for matching, the collaborative filtering relies on the customers' past behavior, and analyzes relationships between users and interdependencies among products to identify new user-item relations. The collaborative filtering systems do not explicitly incorporate feature information, but usually incorporate the information in preference similarity across individuals.

There are two classes of collaborating filtering methods. The first one is call memory-based model, which is also known as instance-based or neighborhood methods. The memory-based models evaluate a customer's preference for a product based on the preference of "neighboring" products by the same user (item-based) or identify like-minded customers who can complement each other's rating (user-based). The memory-based collaborative filtering methods are easy to implement and highly effective. In a report by Amazon research team (Linden et al. 2003), they compare their item oriented CF method with user oriented method,

clustering method and search-based method. The comparison shows that user oriented CF method is impractical on large scale data set, cluster models hurt the recommendation quality and search-based models fail to provide recommendations with interesting, targeted titles, while the item-based collaborative filtering method is efficient and provide high quality recommendations. Therefore, this study will extend the item-based collaborative filtering method.

Another group of methods is call model-based algorithms. These methods compile the available user preferences into compact statistical models from which the recommendations are generated. The most popular ones include singular value decomposition to identify latent structure in ratings (Billsus and Pazzani 1998); probabilistic clustering and Bayesian networks (Breese et al. 1998); dependency networks (Heckerman et al. 2001); latent class models (Hofmann and Puzicha 1999) and latent semantic models (Hofmann 2004) to cluster the ratings; and flexible mixture models to separately cluster users and items (Si and Jin 2003). Unlike the instance based approach, the model-based algorithms are slow to train, but once trained, they can generate recommendations quickly.

With the success of the Netflix competition (Koren et al. 2009), the matrix factorization models are getting great attention. The matrix factorization model belongs to the latent factor models. The latent factor models try to explain the ratings (or preference) by characterizing both products and consumers on factors inferred from ratings patterns. For products, each factor measures the item's characteristic in that dimension. And for customers, each factor denotes how much the customer likes the product on the corresponding attribute. Based on that, the matching between products and customers would equal the dot product of the product's and customer's factor vectors. The most successful

realization of latent factor models is based on matrix factorization. The matrix factorization is a technique for dimension reduction. Besides the explicit rating information, the matrix factorization model allows incorporation of addition information, like implicit feedback including customers' purchasing history, browsing history, search patterns. The extension of the matrix factorization model includes adding item/user biases and considering temporal dynamics (item's popularity changing over time and users' baseline changing over time).

The fundamental assumption of the collaborative filtering method is that if users *X* and *Y* rate *n* items similarly, or have similar behaviors (e.g., buying, watching, listening), they will rate or act on other items similarly. Then we could use the explicit (eg. rating or reviews) or implicit (e.g. browsing or purchasing behavior) information to collaborate the user's preference and then make recommendations based on that. However, users' need may change over time, and this phenomenon is quite often when the customer's shopping is heavily driven by the task or project they are working on. In this scenario, the user-item based collaborating will not work well since what customers have purchased may share low similarity with future purchase. In this study, we will extend the user/item-based collaborative filtering method to itemset-based collaborative filtering. The assumption is that the products purchased can reflect the need or preference relationship between the on-going project and the products set. Although we could not directly identify the on-going project, we can partially recover the project's information from the frequent purchased products-sets. And the proposed method will generate more reliable project-oriented recommendations.

Association Rule Mining

The association rule mining is firstly proposed by Agrawal et al. (1994), which initializes the field of association rule mining. After that, researchers endeavor a lot to improve the efficiency and accuracy of the Apriori algorithm and also develop some other

algorithms (e.g. FP-tree by Han et al. 2000, PRICES algorithm by Wang and Tjortjis 2004) to enhance the original design. But the detail of the algorithms is not what we will discuss in this study. The passing decades have witnessed this method becoming an interesting and well-established area, which studies the co-occurrence relations and patterns among variables/items in large databases. At the beginning, this method is mainly used in the area of market basket analysis in business. After years' development, this method has been applied in broader areas including crime pattern mining (Usha and Rameshkumar, 2014), disease symptom predication (Mocormick et al. 2011), fraud detection (Phua et al. 2010), etc.

To generate interesting rules to guide the business, various measures have been developed, including *support*, *confidence*, *lift*, *conviction*, etc. Lallich et al. (2007) draw a summary of the association rule interestingness measures. They argue that researchers or practioners have to choose the measures best suited to the problem, not limited to the Piatetsky-Shapiro schema (using the support and confidence), and validate the interesting rules against the measures. The measures themselves could only reflect partial information in the dataset, so more metrics need to be developed to recover the full information related to the frequent patterns.

Above that, there are a lot variations and extensions on the market basket analysis. Generalized association rule mining with product category information can discover more useful knowledge by taking application specific information into account (Thomas and Sarawagi 1998). Spatial database (Koperski and Han 1995) also includes the location or geographic information. Temporal association rule mining takes the time of the itemsets into consideration by adding time marks for each transaction line (Agrawal and Srikant 1995).

Besides the traditional mining on the Boolean attribute values, there are a stream of studies discussing the quantitative association rule mining.

Shopping basket analysis produces the best results when the items occur in roughly the same number of transactions in the data. This helps prevent rules from being dominated by the most common items. Also for companies having a rich product variety, the association rules generated at the product level may be of less interests. Therefore, the product hierarchical structure can help here. By rolling up rare items to higher product cateogry levels in the hierarchy, the rules will become more frequent. More common items may not have to be rolled up at all. Nevertheless, in the hierarchical higher level, still some segment/line dominate. There are ways to deal with the problem with uniform support/confidence using group based measure or reduced min support at lower levels transaction (Han et al. 2011).

Association Rule-based Recommendation

In this study, we will not focus on how to efficiently retrieve association rules from the transaction data set. The goal is to discuss how we should utilize the association rules to generate high quality recommendations for customers and empower their purchasing decision-making.

The Association Rule based collaborative filtering algorithms are more often used for top-N recommendation tasks than prediction ones. Sarwar et al. (2000) describes their approach to using a traditional association rule mining algorithms to find rules for developing top-N recommender systems. They find the top-N items by simply choosing all the rules that meet the thresholds for support and confidence values, sorting items according to the confidence of the rules so that items predicted by the rules that have a higher confidence value are ranked higher, and finally selecting the first *N* highest ranked items as the

recommended set. Fu and Han (1995) develop a system to recommend web pages by using a priori algorithm to mine association rules over users' navigation histories. Leung et al. (2006) propose a collaborative filtering framework using fuzzy association rules and multi-level similarity. Other model-based CF techniques include a maximum entropy approach (Pavlov and Pennock, 2002), which clusters the data first, and then in a given cluster uses maximum entropy as an objective function to form a conditional maximal entropy model to make predictions.

Zaïane (2002) propose a method that finds all eligible rules (rules whose antecedents are subsets of the basket and whose consequents are not) and recommends the consequent of the eligible rule with the highest confidence. Wang and Shao (2004) suggested considering only maximal rules, i.e., eligible rules whose antecedents are maximal-matching subsets of the basket. All these approaches focus on identifying a single rule to make the recommendation. The recommendation is made on the partial information—items not present in the maximal rules used are ignored. The set of eligible rules often contains multiple rules with the same consequent, and the quality of recommendations could improve by combining such rules effectively.

The notion of combining rules has been explored in a few studies in the past. Given a customer's basket, Lin et al. (2002) calculate the score for each item as the sum of the products of the supports and confidences of all eligible rules with that item as the consequent. The item with the highest score was recommended to the customer. Wickramaratna et al. (2009) present an approach to identify rules that predict the presence and absence of an item, and proposed a Dempster–Shaffer-based approach for combining rules when some rules predict that a customer will purchase an item, whereas other rules predict the contrary.

However, they noted that their approach is not scalable for real-time applications. Ghoshal et al. (2015) proposed an algorithm to search for the admissible group for the predicted item, and recommend the item which could give the highest mutual information value. Although these methods all try to improve by combing more eligible rules, Liu's method consider all eligible rules without any weights to measure the match between the basket and the eligible rules, so this method will add a lot noise and lead to low quality recommendation. Ghoshal's proposed method has a strong restriction that the admissible group is composed of disjoint rules, which lead to a loss of information from other related rules.

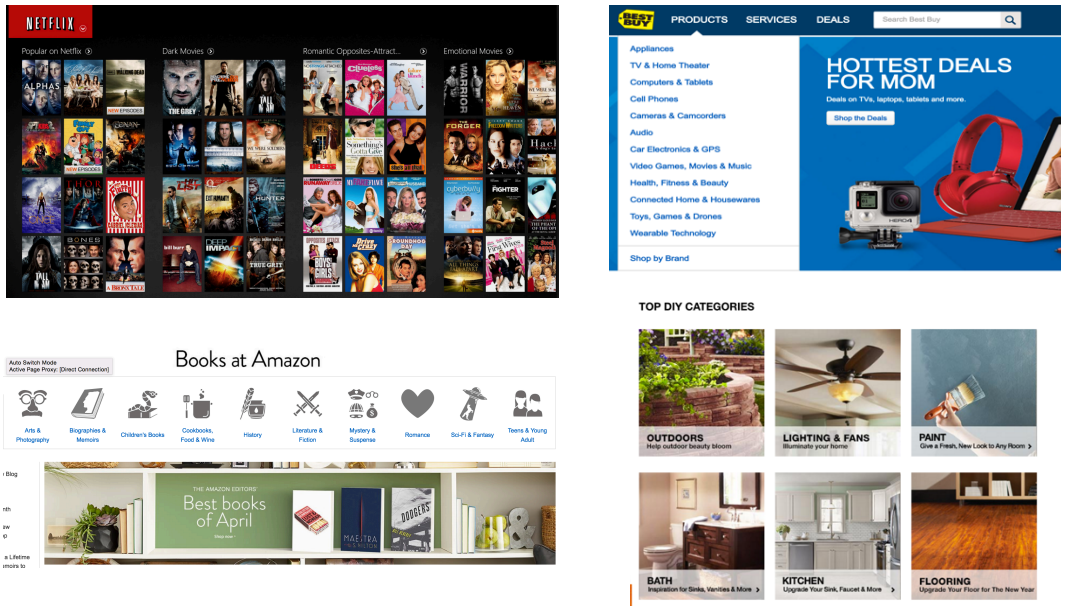
In real world, the data sparsity could be a big challenge for utilizing the rules. For a company with a rich product variety, the frequent product purchasing patterns may not cover all products. It's highly possible that given the basket, we could not find enough eligible rules for the recommendation generation. Ziegler et al. (2004) propose a hybrid collaborative filtering approach to exploit bulk taxonomic information designed for exact product classification to address the data sparsity problem of collaborative filtering recommendations, based on the generation of profiles via inference of super-topic score and topic diversification. The relationships between super-concepts and sub-concepts provide powerful inference opportunities for profile generation based upon the classification of products that customers have chosen.

Based on the literature, we are interested in using the retrieved association rules to make project-oriented product recommendation. The basic assumption in this study is the occurrence of the frequent products patterns is largely driven by the on-going projects. This study is trying to make recommendations for customers based on the match between the current shopping basket and the frequent product patterns. In addition, we make use of the

product categorical information as a way to deal with the data sparsity problem and measure the similarity between products-sets.

Problem Description

Based on the literature, most of the existing studies developed recommender systems using explicit information. Even if using implicit information from the customers' browsing or purchasing history, the products they studied were either movies or book. This is because these products belong to the same functionality category and the customers' preferences or tastes play a key role in determining their purchasing decisions. For these products, we could infer customers' preference based on their transaction record and use this preference information to predict the future purchasing decisions. So we classify these products into topic-driven purchasing category. Music, movies, and books, etc. could be classified in this category.



Topic-Driven Project-Driven

Figure 2.1 Topic-Driven Purchase versus Project-Driven Purchase

Another group of products, which is driven by the function or task, can be taken as project-driven products. For the appliance and electronic products, the customer may work on building up a home theatre, so his/her purchase may be related to DVD players, speakers and cables. But later next month, the customer becomes interested in updating her desktop. Then the transaction in the following month could be hard drives, monitors and so on. In this project-driven scenario, the products bought for the former project would have very low similarity with the products for the later one. Thus we cannot rely the transaction data to infer customers' preference and base on the preference to make recommendations.

In this study, we propose an itemset-based collaborative filtering methods. Although the products set purchased by the same user across time may share low similarity, the products bundles bought by different customers working on the same project will be highly similar with each other. The item-set based method means we could match the products in the basket or already purchased with those in the frequent patterns, based on which we could tell whether the focal customer is working on some project that is similar with the frequent products bundles. Then we could make reliable recommendations for the customer.

Notations

The shopping basket and the item-set in the association rule mining is denoted as $\{A1, B1, D1, F1\}$, each character here stands for the categorical information of the product in the hierarchical structure. And the subscript number is product's position at the product leaf level in the category structure. The character and the number together is the product's SKU. The association rule at the product level is composed by the antecedent and the consequent: $\{A1, B1, D1, F1\} \rightarrow \{X2\}$.

Companies would always keep a record of their historical transaction data, which includes product SKU, customer ID, transaction date and so on. For recommendation, we

could use the product SKU and customer ID information with collaborative filtering techniques to make predictions. Most firms would have a category structure to manage their product line, so here we take the product taxonomy structure from Han et al. (1999) to encode the product structures (in Figure 2.2).

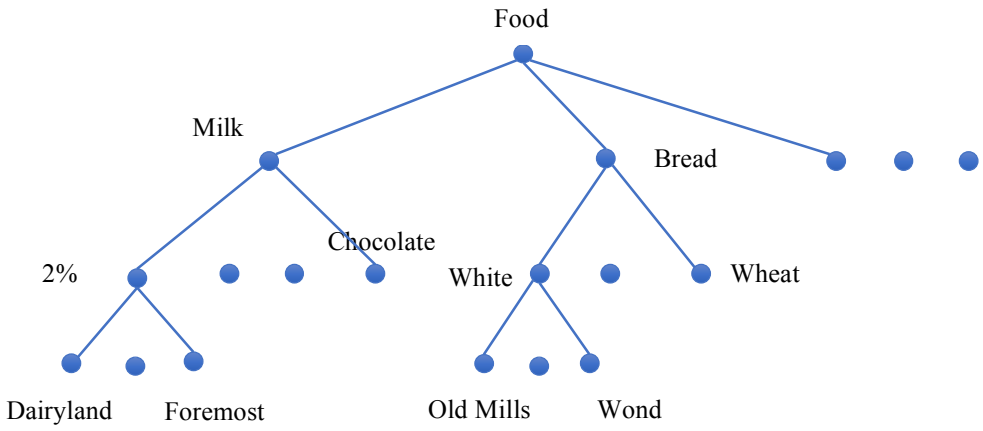


Figure 2.2 Product Category Structure

For example, the Old Mills Whilte bread belong under the Food category, so it could be encoded as [Food, Bread, White, Old Mills], and in practice for simplicity this can be transformed as a vector [1,2,1,1], similarly the Dairyland 2% Milk can be encoded as [1,1,1,1]. Each element in the encoded vector stands for which class/category the product belongs to at that product category level. And for a higher subcategory, the chocolate is encoded as [1,1,4, *], where the unknown leaf is replaced with *.

Project-driven Transactions

In this study, we'd like to analyze the interesting patterns from the project-driven transactions. The most ideal case is each shopping basket is composed by products for one project.

However, most customers may not have a perfect shopping list when they prepare for the project. The complexity of the problem is that the customer may purchase products related to one project in sequential shopping trips. As shown in the following Figure 2.3, the first scenario is that the customer is not an expert in doing the project. It's highly possible that in the first shopping trip, he/she doesn't get all what he/she actually needs. Then the customer need to make consecutive shopping trips to make up for the missing products they need for the project.

Another possibility is that the customer may get involved in different projects during a period of time. For example, when a customer buys a new house, he/she may need to redecorate the house by replacing the floor board and paint the wall. Thus in one shopping transaction, the customer may purchase products related to both projects.

And the hybrid of the two scenario mentioned above is possible when the customer buys products for several on-going projects in consecutive shopping trips.

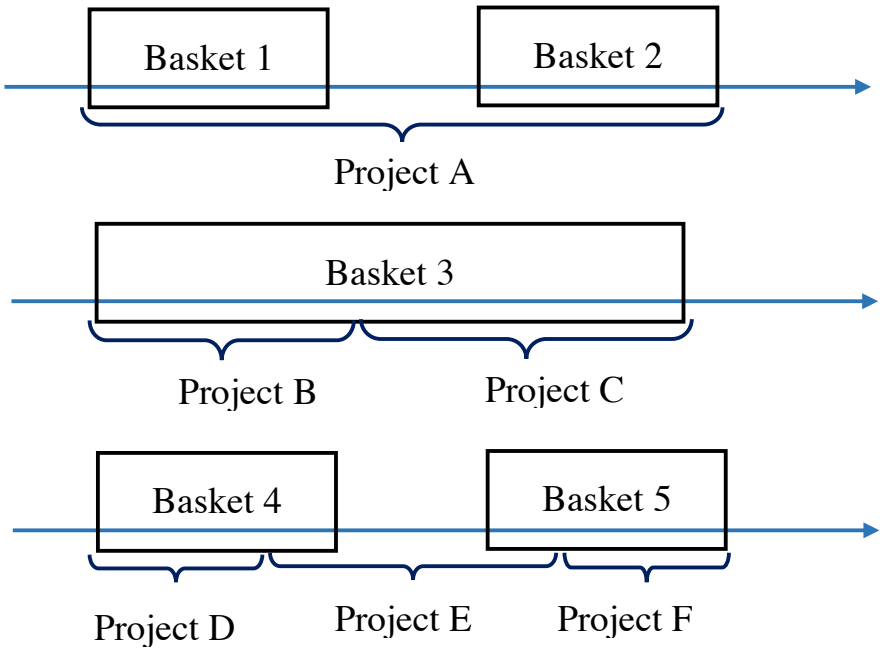


Figure 2.3 Project-Driven Shopping Transactions

Therefore, it is hard to identify the on-going projects from the raw transaction records. However, when there are large amounts of customers' transaction records, some products-sets (core products in the project) will occur frequently in the dataset.

In developing recommendation system, we need to try to identify what project(s) the customer is currently working on based on the products in the shopping cart, and then recommend the most likely useful product (for the related project) to the customer. The complexity of the multiple trips with multiple projects leads to fragments of the products sets for one project falls into multiple frequent item set (in multiple association rules). Hence, the probability we can identify what exactly is the on-going project is very low. However, the association rule mining method could help partially recover the partial information of the project. By comparing the existing shopping basket with the frequent pattern fragments, we could identify a proportion of the on-going project. Then using similar neighbor methods from collaborative filtering, we could provide reliable recommendations for the customer.

Data Sparsity Challenge

Another hurdle in item recommendation is some products may be functionally substitutive to each other, so for the same project, different customers may select different products according to their preference or brand loyalty. When the company has a rich product lines, there will be large number of substitutive products for the same function. If one functional category has thousands of products (varying in size, versions, or brands, etc.) serving the same function, there is a great chance that each single product may be purchased less frequently. Therefore, the frequent products set (association rules) we get at the product level will be of very low support. Such information would be ignored if we set a high threshold for support or confidence.

However, when given the product category information, we may find that product in the same level subcategory would be highly substitutive to each other. Like the product category tree structure in Figure 2.2, Dairyland 2% milk is highly substitutive to Foremos 2% milk.

Given the shopping basket $\{A1, B1, C1\}$, we have to recommend products the customer may need. Assume in the association rules set, we have $\{A1, B1, C1\} \rightarrow \{D1\}$ and $\{A1, B2, C1\} \rightarrow \{D1\}$. Current methods would just take the information from the first rule to predict the related products, since $\{A1, B1, C1\}$ is a perfect maximal eligible rule for the basket. However, the second rule also give some "support" based on the basket, because item $B2$ and $B1$ belong to the same subcategory and they may perform the same function in the project. For example, customers who are working on their garden have purchased spade, rose seeds and fertilizer would purchase water pipe. Another group of customers may buy spade, tulip seeds, fertilizer and water pipe together. If the given shopping basket has spade, rose seeds and fertilizer included, considering both frequent patterns may be helpful for the prediction, although the second pattern is not exactly an "eligible" subset of the basket.

Association Rule-based Recommendation

Companies always keep a record of the customers' historic transaction data. Based on the transaction records, the association rule mining method could generate interesting shopping pattern. The task of the itemset based recommendation system is given the shopping basket, $\{A1, B1, C1\}$, how to utilize the information from the rules and predict which product(s) will be added into the shopping cart.

There is prior knowledge about the product category structure, like product B1 and product $B2$ belong to the same subcategory and are substitutive to each other.

Table 2.1 Available Association Rules

Rule	**Antecedent**	**Consequent**	**Support**	**Confidence**
R1	{A1, B1 }	{D1}	0.02	0.89
R2	{A1, B2}	{D1}	0.06	0.71
R3	{C1}	{D1}	0.08	0.65
R4	{B1, C1}	{D2}	0.025	0.91
R5	{B2, C1}	{D2}	0.08	0.78
R6	{A1 }	{D2}	0.09	0.58

In brief, the problem is that given the association rules and the products current in the shopping cart, how to measure the similarity between the basket and the rules and use the information from rules to generate high quality recommendations.

In Ghoshal et al. (2015), the basket is decomposed into admissible groups and select the admissible group that gives the largest mutual information. For example, given the information in the table above, we have the admissible group $\{A1, B1\}$ and $\{C1\}$ recommending $\{D1\}$, and another group $\{B1, C1\}$ and $\{A1\}$ referring to $\{D2\}$. By comparing the two groups' mutual information, we could decide whether to recommend $\{D1\}$ or $\{D2\}$.

However, in this study, we want to extend the admissible group method by taking the substitutive information from the product category. Although R2 in Table 2.1 may not directly contribute to the reliability of the first admissible group (not an exact subset of the basket), when B1 and B2 are substitutive to each other, the R2 would support the recommendation. And R5 would support the second recommendation choice. It is possible

that taking this information into consideration would change the order of the recommendation list.

Project-oriented Rule-based Recommendation

This study aims to predict what the customers will purchase based on the composition of the customer's shopping basket. In the life time of the customer, he/she could only work on one specific project once, therefore, the transaction history may not help in prediction what the customer will work on in the future. So the traditional item-based collaborative filtering method, which makes recommendations from similar products based on the historic transaction records would not work here. Also the item-based method usually utilizes the co-purchase information between two products, while neglecting the joint distribution of products in the set. In this section, we will use the association rules, which capture the co-occurrence of a bunch of products in the transactions, as the prior knowledge for recommendation generation.

Generate Association Rules Set

Since our proposed recommendation algorithm is based on the association rules, the first step is to generate association rules. The Apriori algorithm, FP-tree method and other algorithms only differ in the efficiency or computing cost and memory cost, which will not affect the set of rules we can get.

In traditional association rule mining, usually the threshold of the support and confidence is set by consulting the experts in the industry. By including the product category structure, less frequently purchased products may also contribute to the support of a higher level product sub-category. Therefore, we should set a low threshold for the support in case losing valuable information. For the threshold of the confidence, we could set it to 30% (or

40%, 50%), which means given the antecedent, we can make a correct prediction on the consequent at a 30% (or 40%, 50%) probability.

Similarity between Product Sets

When generate recommendations, the prior knowledge we have is the products in the current shopping basket and the association rules retrieved. To utilize the association rule, we should know how similar or close the frequent patterns in the rules are with the products set in the basket. Or in other words, to what extent the information from the rules can be applied in analyzing the shopping basket.

Similarity Metric

In the literature, the item-based collaborative filtering methods usually transform the transaction records into an item-item co-purchase matrix. In another way, the transaction records could also be transformed into a transaction matrix: each row stands for one transaction, and each column stands for one product/item, and the binary (0-1) denotes whether the product is purchased in the transaction or not. Similarly, the current shopping basket and the antecedent of the association rule can be conceptualized as a product vector.

Since in this study, we try to utilize the product category information in measuring the similarity between products sets, we could construct a product-category matrix, which denotes the "belonging to" relationship between products and the category.

Based on the literature, there are widely used measurements for the similarity between vectors— *Person Correlation* and *Cosine Correlation*. Here we have two product sets $\vec{a} = (r_{a1}, r_{a2}, \dots, r_{aN})$ and $\vec{b} = (r_{b1}, r_{b2}, \dots, r_{bN})$, where N is the number of products.

Pearson Correlation:

$$Corr_{\vec{a},\vec{b}} = \frac{\sum_{i=1}^{N}(r_{ai}-\overline{r_a})(r_{bi}-\overline{r_b})}{\sqrt{\sum_{i=1}^{N}(r_{ai}-\overline{r_a})^2 \sum_{i=1}^{N}(r_{bi}-\overline{r_b})^2}} \quad (2\text{-}1)$$

Cosine Correlation:

$$Cos_{\vec{a},\vec{b}} = \frac{\vec{a}\cdot\vec{b}}{||\vec{a}||*||\vec{b}||} \quad (2\text{-}2)$$

The cosine correlation measures the angle between the two product vectors, which equals the dot product of the two vectors divided by the product of their scales.

In this proposed recommendation methods, we'd like to utilize the product category information and construct the transaction-category vector. For example, given we have the transaction-product matrix, and the product-category structure, we could generate a transaction-category matrix which capture what categories have been purchased in the transaction as follows:

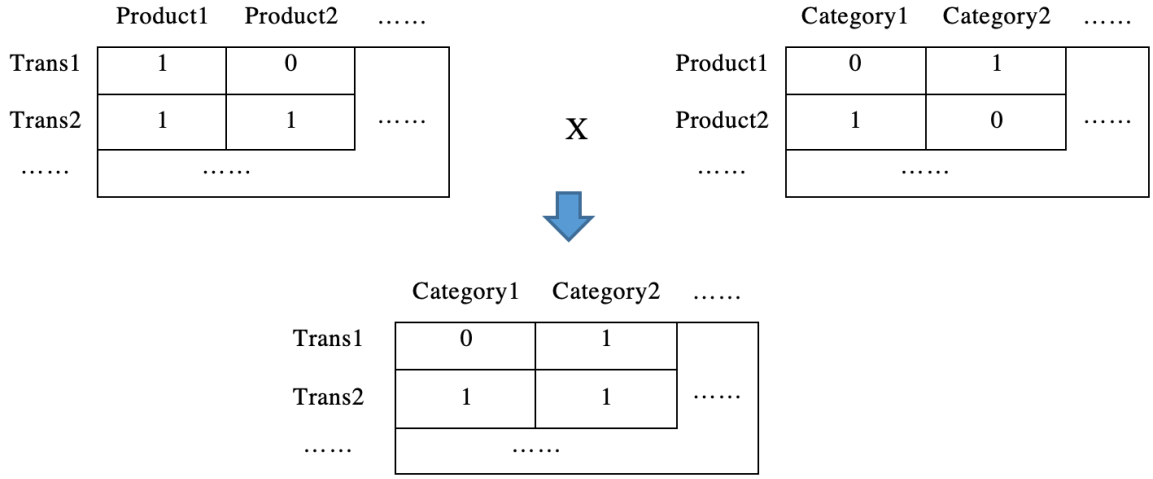


Figure 2.4 Transformation from Transaction-Product Matrix to Transaction-Category Matrix

Based on the transformation above, we can get the transaction-category matrix. And each transaction could be vectored at product category level. For example, if the product

structure has three levels of category: product group, product class and product sub-class. Each transaction/antecedent/basket can be denoted $\overrightarrow{G_a} = (G_{a1}, G_{a2}, \dots, G_{ag})$ (at group level), $\overrightarrow{C_a} = (C_{a1}, C_{a2}, \dots, C_{ac})$ (at class level) and $\overrightarrow{S_a} = (S_{a1}, S_{a2}, \dots, S_{as})$ (at the sub-class level), where *g*, *c*, *s* stand for the number of product groups, classes and sub-classes.

The similarity between two products sets $\vec{a}, \vec{b}$ can be expressed as:

$$Sim(\vec{a}, \vec{b}) = Cos(\overrightarrow{G_a}, \overrightarrow{G_b}) * [\alpha * Cos(\vec{a}, \vec{b}) + \beta * Cos(\overrightarrow{C_a}, \overrightarrow{C_b}) + \gamma * Cos(\overrightarrow{S_a}, \overrightarrow{S_b}) + \cdots] \quad (2\text{-}3)$$

The similarity includes all the cosine similarities at different product category levels: $Cos(\vec{a}, \vec{b})$ measures the cosine similarity at the product vector level, $Cos(\overrightarrow{G_a}, \overrightarrow{G_b})$ is the group level cosine similarity, and so on. Also, we could set different weights (α, β, γ, ...) to different level's similarity. If we need to emphasize exact match of the product, the product similarity weight α can be assigned a large value. When we'd like to emphasize the functionality of the product at the sub-class level, we can give a larger weight on γ.

To reduce the computing cost, the similarity at the product group level is moved outside as an indicator. If the products sets have no similarity at the group level, there is no chance they could share some similarity at lower level.

Recommendation Generation

For recommendation, the system could recommend one product with the highest probability to be purchased or a list of N products with an order by their predicted purchasing probability. In this study, we will combine the information from the basket with that from the association rules, and generate top N related products.

Based on the review of the literature, it shows that confidence is the most popular metric for the ranking of related products. Following the literature, in this study we use the

weighted (based on the similarity between basket and the rules) confidence of the products as a metric to rank.

Given the similarity between the current shopping basket and the retrieved rules, we group the rules by the consequent. For each group, we combine the rules by the similarity we get from previous steps, and normalize it and recalculate the confidence or the product of support and confidence. Then we can get the metric for each recommended product, and we can rank these recommended products.

Experiment

Data Set Description

The data set is from a multi-billion dollars, multi-national specialty retailer. There are over 100,000 different products, and we have a clear product categorical structure. The products are divided into 18 groups, over 200 classes, and almost 2,000 sub-classes. Based on the structure, products performing similar functions are usually divided into one sub-category.

The transaction records are within a two-year time window, which includes a sample of 60,000 customers. For the purpose of CRM (customer relationship management), the company divide the customers into professional customer and regular customer groups. Since the professional customer's purchasing behavior is not exactly driven by the on-going project, in this study, we focus on the project-driven purchasing behavior of the regular customers (about 40,000).

Despite the large size of the dataset, there is still a severe data sparsity problem due to the rich product variety. Each transaction is defined as one unique customer's shopping trip at one store on the same date. Based on this, the dataset has almost 700,000 transactions.

However, among these transaction records, only 76.96% of the products have ever been purchased.

Table 2.2 The Composition of Regular Customer's Shopping Basket

Number of Products in the Basket	**Frequency of Baskets**
1	32.45%
2	22.83%
3~5	30.51%
6~10	11.35%
>10	2.85

The size of the baskets shows that half of the shopping baskets have only one or two products, which shows some evidences that most of the time customers do not have a clear to-buy shopping list so they need make several shopping trips to purchase the products for one or several on-going project(s).

To deal with data sparsity, one way is to combine consecutive transactions of the same customer according to the time sequence. Another way is to use product category structure information. Our first stage attempt is to combine all the customers' shopping records into one transaction for the customer. Although this combination misses the sequential shopping information, the records still contain all the products the customer has purchased in the observation time window,

The summary of the association rules generated from the dataset is shown below in Table 2.3:

Table 2.3 Summary of the Retrieved Association Rules

Support	**Confidence**	**No. of Rules**	**No. of Products Covered**
0.0002	30%	657,359	36,857
0.0002	40%	497,545	36,857
0.0002	50%	389,642	36,857
0.0008	30%	15,854	13,511
0.0008	40%	12,878	13,511
0.0008	50%	10,381	13,511
0.002	30%	1,744	4,906
0.002	40%	1,427	4,906
0.002	50%	1,157	4,906

From Table 2.4, we can tell there is severe data sparsity problem because among the over 100,000 different products, the association rules can only cover a small proportion of them. In practice, there would be a tradeoff in selecting a proper threshold for the support— a higher support will reduce the computing complexity but provide less information (covering a smaller number of products), while a lower support will generate more information in products co-purchasing patterns at the cost of computing burden.

Evaluation of the Performance

In this study, we propose to recommend the top N products based on the ranking computed by the algorithm. To evaluate the performance, we divide the dataset into training set (80%) and testing set (20%), and run a 10-fold cross validation to evaluate the average

performance. Since each transaction record represents one customer, we can randomly sample from the regular customers and separate these customers into training and testing groups.

Since the proposed recommendation method will generate a list N-products, according to literature, we use two measures, precision and recall, to evaluate the recommended result.

The precision measures among all the recommended N products, what proportion eventually get purchased by the customer.

$$Precision = \frac{size\ of\ hit\ set}{size\ of\ the\ top{-}N\ set} = \frac{|test \cap top{-}N|}{N} \quad (2\text{-}4)$$

The recall measures among the real future purchase of the customer, what proportion is correctly predicted by the recommendation algorithm.

$$Recall = \frac{size\ of\ hit\ set}{size\ of\ test\ set} = \frac{|test \cap top{-}N|}{|test|} \quad (2\text{-}5)$$

Based on these two measures, we can evaluate how good the proposed recommendation algorithm performs.

Since the recall is affected by the size of the future basket, it's hard to predict. Thus precision is more important here as it shows among the top N products we have recommended, how many actually get purchased.

Empirical Results

The recommendation result is shown below in Table 2.4.

The higher the support, the rule will only cover a smaller proportion of products. When support is increased, there is no decline trend in precision and recall. On the opposite, for the same confidence level, the higher the support, the precision and recall are increased.

Given the threshold of support, the higher confidence in general leads to a lower precision and recall values, which is because a higher confidence value will cut the number of rules retrieved and provide less information to absorb. The exception is that for support at 0.002, when the confidence increases from 40% to 50%, the precision and recall unexpectedly increases.

Table 2.4 Top-N Recommendation on Different Support and Confidence

Support	**Confidence**	**top-N**	**Precision**	**Recall**	**Products Covered**
0.0008	30%	10	3.05%	1.50%	13378
0.0008	40%	10	2.94%	1.44%	13378
0.0008	50%	10	2.96%	1.42%	13378
0.002	30%	10	3.03%	1.56%	4903
0.002	40%	10	2.95%	1.52%	4903
0.002	50%	10	3.11%	1.54%	4903
0.01	30%	10	3.49%	1.70%	374
0.01	40%	10	3.38%	1.59%	374
0.01	50%	10	3.29%	1.52%	374

Next, we need to check how the similarity weights on different category levels impact the prediction results. Especially we are interested in whether the product similarity (α) or the sub-class similarity (γ) plays a more important role here. Considering the product

coverage affected by the support and confidence, we set the thresholds for support and confidence for rules selection as: support=0.01 and confidence=80%.

Table 2.5 Comparison Between Product Level and Multi-Level Rules

Top-N	**Product Level Rules**	**Multi-Level Rules**	**Improvement**
N=3	3.47%	3.82%	10.09%***
N=5	2.98%	3.19%	7.0%***
N=10	2.13%	2.14%	0.47%

When α =1, it means we only consider the similarity at the product level. The second column in Table 2.5 shows the performance of recommendation based on rules only learned from the product level. The third column demonstrates the performance of multi-level rules based recommendation, with 30% weights from product level, 10% from product class category level, and the rest 60% relies on the sub-class category level. Overall, it shows that utilizing information from product class and sub-class categorical structure could help improve the performance of association-rule-based recommendations. The improvements in precision are significant.

Conclusions

In this study, we are interested in discovering the hidden on-going projects from the customers' historical shopping basket information. And the association rule mining method has been used in the potential project identification. To get more meaningful potential

project, we will adapt the multi-level association rule mining methods and sequential rule mining algorithms.

We believe that the proposed method can add to the existing literature on association rule mining and inspire further research on the identification of more complex patterns in market basket analysis. The practical contribution of the proposed research is that it can help companies better understand customers' project related purchase behaviors and better predict future demands of its products. In a longer term, the proposed method could be applied in other related domains as well.

There are also some interesting future research directions worth exploration. First of all, there is a lack of theoretical foundation for rule-based recommendations. Some model developments based on statistical theories would be promising. Secondly, a context-aware recommendation system would help firms utilize available context related information to improve the performance of recommendation systems. For instance, the seasonal indicator can be incorporated and whether the customer is a business buyer or a regular consumer may direct the recommendation algorithm to more accurate predictions.

References

Agrawal, R., Imieliński, T., Swami, A. 1993. "Mining Association Rules between sets of Items in Large Databases," In *ACM SIGMOD Record*, 22(2), pp. 207-216.

Agrawal, R., and Srikant, R. 1994. "Fast Algorithms for Mining Association Rules. Proc. 20th int. conf. very large databases," *VLDB*. Vol. 1215.

Agrawal, R., and Srikant, R. 1995. "Mining Sequential Patterns. In Data Engineering," In *Proceedings of the Eleventh International Conference*, pp. 3-14.

Agrawal, R. C., Aggarwal, C. and Prasad, V. 2000. "A Tree Projection Algorithm for Generation of Frequent Item Sets," *Journal of parallel and Distributed Computing*, 61(3), pp 350-371.

Askar, P. 2016. “Beyond Market Basket Analysis: Extending Association Rules,” Ironside, Downloaded from https://www.ironsidegroup.com/2016/01/11/beyond-market-basket-analysis-extending-association-rules/.

Asthana, P., Singh, A. and Singh, D. 2013. “A Survey on Association Rule Mining Using Apriori Based Algorithm and Hash Based Methods,” *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7), pp. 599-603.

Billsus, D. and Pazzani, M.J. 1998. “Learning Collaborative Information Filters,” In *ICML*, 98, pp. 46-54.

Blanchard, D. 2007. “Supply Chains Also Work in Reverse,” Downloaded from Industry Week. http://www.industryweek.com/planning-amp-forecasting/supply-chains-also-work-reverse.

Blattberg, R. C., Kim, B., and Neslin, S. A. 2008 *Database Marketing- Analyzing and Managing Customers* (Springer).

Breese, J. S., Heckerman D., and Kadie C. 1998. “Empirical analysis of predictive algorithms for collaborative filtering,” In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43-52. Morgan Kaufmann Publishers Inc.

Chuang, K., Chen, and M., Yang, W. 2005. “Progressive Sampling for Association Rules Based on Sampling Error Estimation,” *Lecture Notes in Computer Science*, Volume 3518, pp 505 – 515.

Fu, Y., and Han, J. 1995. “Meta-Rule-Guided Mining of Association Rules in Relational Databases,” In *KDOOD/TDOOD*, pp. 39-46.

Ghoshal, A., Menon, S., and Sarkar, S. 2015. “Recommendations Using Information from Multiple Rules,” *Information Systems Research*, 26(3), pp. 532–551.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), pp. 61-70.

Gordon, L. 2008. “Leading Practices in Market Basket Analysis,” White paper, Fact Point Group. Downloaded from http://www.irgintl.com/pdf2/1.pdf.

Han, J., and Fu, Y. 1995. “Discovery of Multiple-level Association Rules from Large Databases,” *VLDB*. Vol. 95, pp. 420-431.

Han, J., and Fu, Y. 1999. “Mining Multiple-level Association Rules in Large Databases,” *IEEE Transactions on Knowledge and Data Engineering*, 11(5), pp 798-805.

Han, J., Pei, J. and Yin Y. 2000. “Mining Frequent Patterns without Candidate Generation,” In *ACM SIGMOD Record*, 29(2), pp. 1-12.

Han, J., Kamber, M., and Pei, J. 2011. *Data Mining: Concepts and Techniques*. Elsevier.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. 2001. "Dependency networks for inference, collaborative filtering, and data visualization," *The Journal of Machine Learning Research* 1, pp. 49-75.

Hofmann, T., and Puzicha, J. 1999. "Latent class models for collaborative filtering," In *IJCAI*, 99, pp. 688-693.

Hofmann, T. 2004. "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS),* 22(1), pp. 89-115.

Hosanagar, K, Fleder, D. M., Lee, D, and Buja, A 2014. "Will the global village fracture into tribes: Recommender systems and their effects on consumers," *Management Science* 60(4), pp.805–823.

Koperski, K. and Han, J. 1995. "Discovery of Spatial Association Rules in Geographic Information Databases," In *Advances in spatial databases*, Springer Berlin Heidelberg, pp. 47-66.

Koren, Y., Bell, R., and Volinsky, C. 2009. "Matrix factorization techniques for recommender systems," *Computer* (8), pp. 30-37.

Kotsiantis, S., and Kanellopoulos, D. 2006. "Association Rules Mining: A Recent Overview," *GESTS International Transactions on Computer Science and Engineering* 32(1) pp 71-82.

Lallich, S., Teytaud, O. and Prudhomme, E. 2007. "Association Rule Interestingness: Measure and Statistical Validation," In *Quality Measures in Data Mining*. Springer Berlin Heidelberg, pp 251-275.

Lin, W., Alvarez, S. A., and Ruiz, C. 2002. "Efficient adaptive-support association rule mining for recommender systems," *Data Mining and Knowledge Discovery*, 6(1), pp. 83-105.

Linden, G., Smith, B., and York, J. 2003. "Amazon.com recommendations: Item-to-item collaborative filtering", *IEEE on Internet Computing*, 7(1), pp. 76-80.

Leung, C.W.K., Chan, S.C.F. and Chung, F.L., 2006. "A collaborative filtering framework based on fuzzy association rules and multiple-level similarity," *Knowledge and Information Systems*, 10(3), pp.357-381.

McCormick, T., Rudin, C. and Madigan, D. 2011. "A Hierarchical Model for Association Rule Mining of Sequential Events: An Approach to Automated Medical Symptom Prediction," *MIT Sloan Research Paper*. Available at SSRN: http://ssrn.com/abstract=1736062.

Pavlov, D. Y., and Pennock, D. M. 2002. "A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains," In *Advances in neural information processing systems*, pp. 1441-1448. Chicago.

Petersen, J. A., and Kumar, V. 2015. "Perceived Risk, Product Returns, and Optimal Resource Allocation: Evidence from a Field Experiment," *Journal of Marketing Research* 52(2), pp. 268-285.

Phua, C., Lee, V., Smith, K., and Gayler, R. 2010. "A Comprehensive Survey of Data Mining-based Fraud Detection Research," arXiv preprint arXiv:1009.6119. Avarilable at SSRN: http://arxiv.org/abs/1009.6119

Priyanka, Er. V., and Sharma, K. 2014. "Apriori Algorithm for Mining Frequent Itemsets-A Review," *International Journal of Computer Application and Engineering Technology*, 3 (3), pp 232-236.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000. "Analysis of recommendation algorithms for e-commerce," In P*roceedings of the 2nd ACM conference on Electronic commerce*, pp. 158-167.

Short, J. E., Bohn, R. E., and Baru, C. 2011. "How Much Information? 2010 Report on Enterprise Server Information," *UCSD Global Information Industry Center*.

Si, L., and Jin, R. 2003. "Flexible mixture model for collaborative filtering," In *ICML*, 3, pp. 704-711.

Srikant, R., and Agrawal, R. 1995. "Mining Generalized Association Rules," *IBM Research Division*.

Tang, P., and Turkia, M. P. 2006. "Parallelizing Frequent Itemset Mining with FP-Trees," *Computers and Their Applications*, pp 30-35.

Thomas, S. and Sarawagi, S. 1998. "Mining Generalized Association Rules and Sequential Patterns Using SQL Queries," In *KDD*, pp 344-348.

Tien D. D., Hui, S. C. and Fong, A. 2003. "Mining Frequent Itemsets with CategoryBased Constraints," *Lecture Notes in Computer Science*, (2843), pp. 76 – 86.

Usha, D. and Rameshkumar, K. 2014. "A Complete Survey on Application of Frequent Pattern Mining and Association Rule Mining on Crime Pattern Mining," *International Journal of Advances in Computer Science and Technology*, 3(4), pp 264 – 275.

Wang, F. H. and Shao, H. M. 2004. "Effective personalized recommendation based on time-framed navigation clustering and association mining," *Expert systems with applications*, 27(3), pp. 365-377.

Wang, C., and Tjortjis, C. 2004. "PRICES: An Efficient Algorithm for Mining Association rules," *Intelligent Data Engineering and Automated Learning–IDEAL*. Springer Berlin Heidelberg, pp 352-358.

Wickramaratna, K., Kubat, M., and Premaratne, K. 2009. "Predicting missing items in shopping carts," *IEEE Transactions on Knowledge and Data Engineering*, 21(7), pp. 985-998.

Yuan, Y., and Huang, T. 2005. "A Matrix Algorithm for Mining Association Rules.," *Advances in Intelligent Computing*. Springer Berlin Heidelberg, pp 370-379.

Zaïane, OR 2002. "Building a recommender agent for e-learning systems," *Proc. Internat. Conf. Comput. Ed*. (IEEE, Washington, DC), pp. 55–59.

Ziegler, C. N., Lausen, G., and Schmidt-Thieme, L. 2004. "Taxonomy-driven computation of product recommendations," In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pp. 406-415.

CHAPTER 3. PREDICTING TIME TO UPGRADE UNDER SUCCESSIVE PRODUCT GENERATIONS

Modified from a manuscript to be submitted to Information Systems Research

Xinxue Qu, Zhengrui Jiang, and Aslan Lotfi

Ivy College of Business, Iowa State University, Ames, IA, USA

Abstract

In the presence of successive product generations, most customers are repeat buyers, who may decide to purchase a future product generation before its release. As a result, after the new product generation enters the market, its sales often show a declining pattern, making traditional bell-shaped diffusion models unsuitable for characterizing the timing of product upgrades by customers. In this study, we propose a survival model with exponential-decay baseline function (or Expo-Decay model) to predict customers' time to upgrade to a new product generation. Compared with existing proportional hazard models, the Expo-Decay model is parsimonious and easy to interpret. In addition, empirical analysis using upgrade and usage data for a major sports video game series shows that the Expo-Decay model performs better than or as well as existing parametric models in prediction accuracy. Furthermore, we extend the baseline Expo-Decay model with the frailty model to incorporate unobservable customer heterogeneity and with the point process method to capture influences of previous adoptions, and with time-variant covariates. Empirical results obtained using the Expo-Decay model reveal that customers' previous adoption and usage patterns can help predict their timing to upgrade to a new product generation. In particular, we find that (i) potential switching customers who have adopted the previous generation are more likely to upgrade; (ii) heavy players tend to upgrade earlier; (iii) specialized customers demonstrate a

lower probability to upgrade. These findings can help firms better understand customers' upgrade behaviors and develop more personalized promotions to target customers.

Keywords: Technology Adoption, Product Upgrade, Video Game, Survival Analysis

Introduction

Continuous product improvement and frequent releases of new product generations is a common practice by firms. Releasing improved product generations enhances customer loyalty and encourages more repeat purchases (Albuquerque and Nevskaya, 2012), thereby increasing sales which otherwise would decrease as a result of market saturation. For example, Call of Duty, the best-selling first-person shooter video game series, releases new game generations every year to blockbuster-level sales. In fact, introducing product improvements may result in a large volume of upgrade sales in a relatively short time period. For instance, on average, 17% of iPhone users upgrade as soon as the new model is released, 58% upgrade one year after the release, and 22% two years after the release. Only 2% of users wait longer (Edwards, 2016).

However, companies' investments in technological improvement may not always lead to popularity of new product generations. If the quality improvement is marginal, customers may be reluctant to upgrade to a new generation. For instance, in recent years, the average time-to-upgrade for smartphones has extended. In 2014, U.S. consumers are upgrading their smartphones every 23 months. Lately, consumers on average are holding onto their phones for an additional eight months. It is estimated the time gap between upgrades will widen to 33 months by 2019 (Martin and FitzGerald, 2018). Therefore, it is important to explore factors that might influence existing users' upgrade intentions.

A limited number of prior studies have focused on factors that might impact customers' future purchases intentions in a product series. To the best of our knowledge, no prior research has examined the influence of consumers' previous adoption and usage experience on their upgrade decisions. This gap is filled by the current study. Furthermore, in the presence of successive product generations, most customers are repeat buyers, who may decide to purchase a future product generation before its release. As a result, after the new product generation enters the market, its sales often show a declining pattern, making traditional bell-shaped diffusion models unsuitable for characterizing the timing of product upgrades by customers. To model this declining sales trend, we propose an Exponential-Decay proportional hazard model (Expo-Decay PHM) to help explain and predict consumers' upgrade behaviors.

Using a rich dataset for a major sports video game series that includes individual-level activation and usage records, we evaluate the proposed Expo-Decay model against existing survival models, and identify predictors of time-to-upgrade decisions. Our results indicate that players' prior adoption and usage experience can indeed help predict their timing of product upgrade. In particular, we find that potential switching customers who have adopted the previous generation are more likely to upgrade to a new generation, heavy players tend to upgrade earlier, and specialized customers have a lower probability to upgrade. Furthermore, empirical results show that the proposed Expo-Decay model performs better than or as well as existing parametric models in prediction accuracy. By integrating the Expo-Decay model with a point process capturing adoptions of previous generation, the prediction accuracy can be further improved.

In the next section, we review existing literature on factors that drive consumers' upgrade intentions. We then briefly introduce the proportional hazard model and propose the Expo-Decay model. Data and experience-based covariates are described afterwards. Empirical estimations and findings are presented next. To incorporate unobservable customer heterogeneity, customers' adoption patterns, and time-variant covariates, extended Expo-Decay models are proposed. We conclude the paper with discussions on main contributions, managerial implications, and future research directions.

Related Literature

Product upgrade can be defined as a user's second or subsequent purchase for an improved version of an earlier product (Kim and Srinivasan, 2009). Though frequent product improvement has become a common strategy for technology companies, customers' upgrade decisions of successive product generations are not well understood. In this section, we will review relevant literature regarding incentives of product upgrade, influence of consumers' previous experience, and duration models for product upgrade.

Incentives of Product Upgrade

Although the TAM (Davis 1986) and the Expectation–Confirmation model (Thong et al., 2006) have been applied to explain technology adoption behaviors, existing theories cannot effectively explain customers' upgrade decisions. In a study that examines factors that may impact customers' upgrades from 2G to 3G mobile phones, Tseng and Lo (2011) find users' satisfaction may hurt their willingness to upgrade to a newer generation and the TAM fails to explain consumers' intentions to upgrade. There is a call for new research frameworks that could help understand consumers' product upgrade intentions.

In innovation diffusion literature, researchers have attempted to identify the influence of consumer characteristics on new product adoption. Based on consumers' innovativeness

and willingness to try new products, Rogers (1995) classifies the buyers into five categories: innovators, early adopters, early majority, late majority, and laggards. The potential adopters' income level, education, occupation, and experience with other related technical products are also found to influence their upgrade propensity toward a new technology (Dickerson and Gentry, 1983). Psychologically, venture-some, impulsive, flexible, and inner-directed innovators are expected to be more open to technology upgrades (Huh and Kim, 2008).

In a multigeneration product series, characteristics of the new generation often create need arousal for upgrades. Van Nes and Cramer (2008) generate a list of product characteristics that motivate replacement: technological performance, hedonic value, features and technological advantages, psychological value, ergonomics, economic value, and ecological benefit. In addition, several moderators may influence consumers' upgrades decisions, such as promotional formats, usage frequency (Okada, 2001), product similarity (Okada, 2006), trade-in conditions (overpaid vs. underpaid for the trade-in, Purohit, 1995), or transaction conditions (buying alone vs. trade-in, Zhu et al., 2008).

Although studies try to identify incentives behind customers' upgrade decisions, factors identified in this research stream are related to the customers' perception of features of the new generation and the marketing efforts, while neglecting the influence of customers' own experience with previously adopted generations. To fill the gap, in this study, we examine the influence of consumers' experience on their future upgrade decisions.

Consumer Experience and Upgrade Decisions

Aside from characteristics of technology improvements and consumers' demographics and psychographics factors, consumers' experience with related technologies are found to impact their upgrade decisions (Dee Dickerson and Gentry, 1983). Consumers familiar with a previous generation are able to utilize their knowledge to learn about a

following generation (Sääksjärvi and Lampinen, 2005) and are more likely to adopt a future generation earlier (Rogers, 2003).

Based on the identified driving factors in literature, Kim et al. (2001) find previous adoption history and post-adoption behavior toward current products are more robust predictors of upgrade decisions. Similarly, Shih and Venkatesh (2004) point out two perspectives for innovation diffusion studies: adoption-diffusion perspective, which is concerned with examining the process through which a target population adopts an innovation, and usage-diffusion perspective, which concentrates on the usage behavior associated with an innovative product. Following the two perspectives, we examine how varying adoption and usage experience of consumers may result in heterogeneous product upgrade decisions.

Previous Adoption Experience and Upgrade Decisions

With the help of widely applied modern data collection and storage technologies, consumers' previous adoption experience become more available. Rijnsoever and Oppewal (2012) show that variables related with previous adoptions outperform conventional socio-demographic and psychographic variables in predicting early adoptions.

Successful adoption experiences with previous generation may positively affect the expectation of possible benefits involved with the product series, therefore reducing resistance against similar technologies (Shih and Venkatesh, 2004; Chang et al., 2005). On the one hand, active engagement in the purchase process will make the consumer more knowledgeable with the product series and various aspects of the purchase process (Alba and Hutchinson, 1987). More importantly, Kameda and Davis (1990) point out the most recent purchase (rather than the entire history) is a good proxy for a reference point for the next purchase. On the other hand, when the customer adopts the current generation in use can

impact her upgrade decision: the longer the time since the last purchase, the stronger the desire for upgrading (Bayus and Gupta, 1992). In addition, the current ownership in the product series also affects early adoption of a new product generation (Rijnsoever and Oppewal, 2012).

Although adopters may have perceived the usefulness or relative advantage of the product series as being high across all versions, they, particularly potential switchers— those who have adopted the recent product generation in the product line, may not perceive the marginal relative advantage of a new generation to be large enough to justify an upgrade. A consumer's motivation to upgrade likely decreases if the version she has already adopted can fulfill her needs (Ellen et al. 1991; Gerlach et al. 2014). Consumers who like certain attributes of the existing products might even negatively react to a substitute that differs on those attributes (Ellen et al. 1991). As a result, how previous adoption experience impacts future upgrade decisions is essentially an empirical question.

Previous Usage Experience and Upgrade Decisions

Users' previous usage experience, in addition to previous adoption experience, provides further insight into their upgrade decisions. Experiences with the product's functionality give the consumer the ability to interpret new innovations and to detect superior new functionalities, leading to a higher likelihood of early adoption (Rijnsoevera and Oppewal, 2012).

Sääksjärvi and Lampinen (2005) first study how usage experience with a previous generation plays a role in perceived risk of adopting a successive generation. Huh and Kim (2008) relate consumers' first adoptions with future upgrade decisions by incorporating their post-adoption usage behavior into the model. Contradicting previous presumptions, while post-adoption usage behavior does impact upgrade decision, they find the usage duration is

not a good predictor of upgrade intention and the behavior toward innovative functions has a significantly stronger impact on upgrade intention than the behavior toward basic functions.

Shih and Venkatesh (2004) conceptualize innovation usage to have two distinct dimensions, variety of use and rate of use, resulting in four distinct usage patterns: intense, specialized, nonspecialized, and limited. They suggest that users demonstrating higher usage patterns are more open to future technologies compared to users showing lower usage pattern.

However, there is a critical research gap in studying the influence of product usage experience on future upgrade decisions, which can be partially attributed to a lack of relevant data. Consumers' usage records are usually not observable to companies or researchers. Following this direction, this study try to fill the gap to fully discover the usage pattern and its impact on long-term purchasing decisions (Golder and Tellis 1997; Shih and Venkatesh 2004).

The present research aims to provide further insight into innovation adoption and post-adoption product usage behavior, and the influences on product upgrade decisions. Based on the rich dataset, we construct our adoption and usage related covariates in Section 4, and examine their influences on upgrade decisions in Section 5.

Time to Upgrade Models

In the literature, discrete choice models have been adapted to explain consumers' upgrade decisions. Kim et al (2001) propose an individual-level multinomial logit model to capture adoption and substitution patterns for successive generations of technological products. Bolton et al. (2008) model a business customer's service upgrade decision as a binary logit model. Aside from the customers' upgrade decisions, understanding the

consumers' purchase timing is also essential since it helps companies better forecast the demand, target potential buyers, promote their products, and manage the distribution.

Most studies in the literature have attempted to model the timing of repeat purchases. The negative binomial distribution (NBD) model is found to provide an excellent fit to repeat purchase data (Ehrenberg, 2004), which assumes that the number of purchases made by a customer in a given time period can be characterized by a Poisson distribution with the buying rate following a gamma distribution. Building on the stochastic counting and timing foundations, researchers have developed a number of models of buyer repeat purchase behavior that make use of data from a firm's transaction databases (e.g. Reinartz and Kumar 2000; Fader et al. 2005).

Only a few of studies have model the purchase timing in the context of high-tech product upgrades, among which the survival model, specifically the proportional hazard model, is the most widely applied. Kim and Srinivasan (2009) propose a conjoint utility model with a hazard function specification examining the upgrade timing of PDAs. Extending from the conventional duration model, Sinha and Chandrashekaran (1992) first develop the split hazard model for the analysis of diffusion of innovation, in which the splitting model indicates whether a customer will eventually adopt the product while the hazard part model the distribution of time to adopt. Prins and Verhoef (2007) apply the split-hazard model to study the effect of marketing communication on existing customers' adoption timing of a new E-service. However, existing models have only considered the observed heterogeneity. In this study, following the survival analysis framework, we propose an Expo-Decay proportional hazard model to study the impact of existing customers' adoption and usage experience on the timing of upgrade. Based on the baseline model, we

also consider some extensions to include the unobservable heterogeneity, the complete adoption history, and time-variant covariates.

Time-to-Upgrade Model Development

Companies often release successive product generations based on a predetermined schedule. For instance, new generations of video game *Call of Duty* are usually released at late October or early November every year. The present study focuses on this type of product series, for which the product series is not new to the market and the release dates of past and future generations are considered public knowledge.

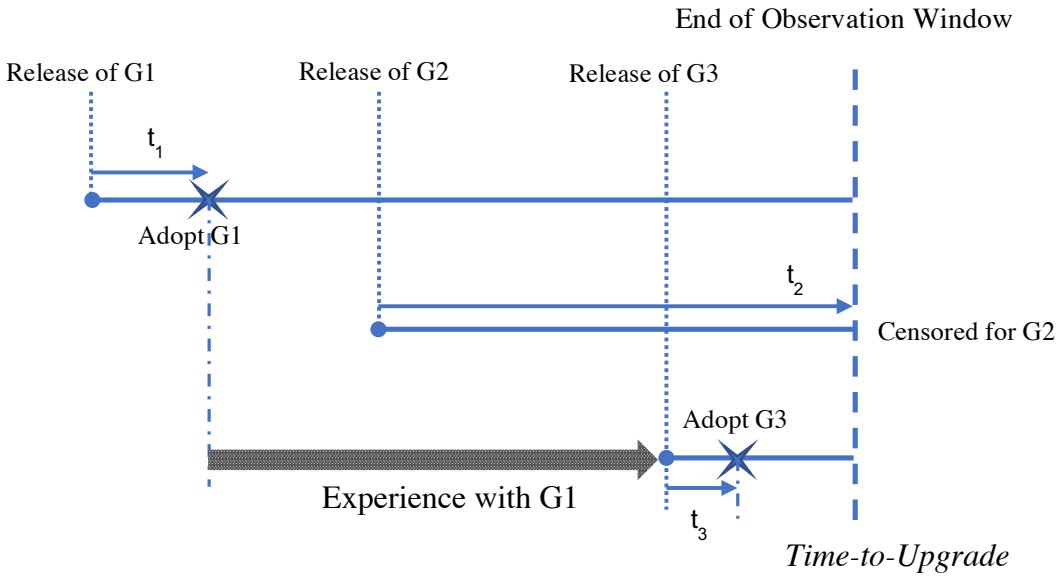


Figure 3.1 An Illustrative Example of Cross-Generation Adoption

As illustrated in the Figure 3.1, at around the same time each year, the company releases an improved generation of its product series. Customers may or may not upgrade to the newest generation every year. The example customer illustrated in Figure 3.1 adopted the first generation (G1) *t1* days after its release, but did not upgrade after the second generation (G2) became available. Now, after the third generation (G3) is launched, the customer

decides whether to upgrade to G3, or continue to use G1 and wait for a further improved future generation.

Based on the theoretical discussions in the previous section, the decision to upgrade to a new generation depends on many drivers. In the present study, we focus on factors that reflect customers' previous adoption behaviors and usage patterns. In this section, we briefly review the survival analysis method and propose an Expo-Decay proportional hazard model to examine how customers' experiences impact their timing of upgrade purchases.

Proportional Hazard Model

Time-to-event survival analysis has been widely applied in business research to model the time duration between customers' repeat purchases (Gupta, 1991; Seetharaman and Chintagunta 2003). Bardhan et al. (2014) extend the proportional hazard model to predict the propensity, frequency, and timing of readmissions of patients.

In our case, a proportional hazard model specification (Cox 1972; Gupta 1991) is applied to explain and predict customers' time-to-upgrade decisions, since it can incorporate the influence of covariates of interest and provide better interpretability. The baseline hazard rate, defined as the (instantaneous) probability of upgrade during an indefinitely small time interval $(t, t + \Delta t)$ conditional on no upgrade having occurred before time *t*, is

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t<T<t+\Delta t | T>t)}{\Delta t} = \frac{f(t)}{S(t)} = \frac{f(t)}{1-F(t)} \quad (3\text{-}1)$$

where $f(t)$ and $F(t)$ are the probability density function and cumulative density function of the distribution of the timing of an upgrade. The survival function $S(t)$ represents the probability that a customer has not upgraded till time t. A survival process can be characterized by the hazard function, the probability density function, or the survival function.

The influences of covariates of interest on time to upgrade can be incorporated into the customer specific hazard rate as:

$$h(t, X_i) = h_0(t)e^{X_i'\boldsymbol{\beta}} \tag{3-2}$$

in which $h_0(t)$ is the baseline upgrade hazard, X_i is a vector of covariates representing customer *i*'s previous adoption and usage experience, and coefficients β capture the impact of these covariates on the time-to-upgrade decision. Incorporating these experience-related covariates, the survival function becomes $S(t,\ X_i) = [S_0(t)]^{\exp(X_i'\cdot\beta)}$ and the probability density function of time-to-upgrade is $f(t, X_i) = h_0(t)e^{X_i'\beta}[S_0(t)]^{\exp(X_i'\cdot\beta)}$.

Common Baseline Hazard Functions

In a proportional hazard model, $h_0(t)$ describes how the upgrade hazard rate changes over time in the absence of influences from related covariates. It is not always necessary to explicitly specify a baseline hazard function in survival analysis, in which case the parameters can be estimated using non-parametric PH models (e.g. piecewise model) or semi-parametric PH model (a.k.a. Cox Model). In this study, we use Cox model as one of the benchmark estimation models and do not consider non-parametric methods because of a larger number of parameters to be estimated and the risk of over-fitting.

Regarding parametric PH models, there are a few widely applied baseline hazard specifications, including Exponential, Weibull, Erlang-2, and Expo-power. The related hazard functions and survival functions are summarized in Table 3.1. The exponential hazard assumes a constant hazard rate, which is a special case of Weibull function. The Weibull hazard can capture constant, monotonically increasing and monotonically decreasing hazard rates, and is the most widely applied in the PHM literature. The Erlang-2 baseline hazard has a monotonically increasing shape and has been widely used in estimating customers' inter-

purchase time distribution. In a comparison study, Seetharaman and Chintagunta (2003) review alternative specifications of the proportional hazard model and find that the flexible Expo-Power specification fits and validates data the best. In this study, the widely applied Weibull model and the most flexible Expo-Power model are used as benchmark methods.

Table 3.1 Parametric Baseline Hazard Functions

	Exponential	**Weibull**	**Erlang-2**	**Expo-Power**
$\boldsymbol{h_0(t)}$	γ	$\gamma\alpha(\gamma t)^{\alpha-1}$	$\frac{\gamma^2 t}{1+\gamma t}$	$\gamma\alpha t^{\alpha-1}e^{\theta t^{\alpha}}$
$\boldsymbol{S_0(t)}$	$e^{-\gamma t}$	$e^{-(\gamma t)^{\alpha}}$	$(1+\gamma t)e^{-\gamma t}$	$e^{\frac{\gamma}{\theta}[1-e^{\theta t^{\alpha}}]}$
Shape of Baseline Hazard	*Flat*	*Flat, monotonically increasing, monotonically decreasing*	*Monotonically increasing*	*Flat, monotonically increasing, monotonically decreasing, U-shaped, or inverted U-shaped*

Pre-release Virtual Adoption

Prior to the release of a new product generation, companies usually advertise it through various channels. For instance, official trailers of the next generation video games will be posted on Youtube.com months before the release date, and short demo-version games can be available for download on platforms a few weeks before the game launch date. More importantly, existing customers, the majority adopters of a new generation, should be well aware of an upcoming new generation even if they are not exposed to such

advertisements. As a result, most potential customers become well aware of the possible release of a future generation, and have reasonable prior expectations about the quality of new product generations.

Due to the pre-release product awareness and information diffusion, potential consumers who anticipate a forthcoming new product generation may commit to buy it prior to its release. In fact, empirical evidences show that the pre-release word-of-mouth (WOM) dynamics can serve as early indicators of future product sales, and products with higher spikes in pre-release WOM tend to have higher initial sales (Gelper et al. 2015). Hence, consumers' upgrade decisions might have been made even before the release of the new product generation, which we refer to as "virtual adoptions."

Despite the pre-release virtual adoptions, actual sales or activations can only take place after the release of the new product generation. Therefore, when pre-release virtual adoptions account for the majority of the product upgrade sales, accumulated virtual adoptions will result in a high upgrade hazard rate in a short time-frame immediately following the product launch, as evidenced by long waiting lines following the release of a new iPhone generation (Nick 2014). For customers who have not upgraded to the new product generation at early stages, their chance to upgrade later will get lower over time, which means a declining hazard rate. As a result, the traditional product diffusion model and the associated bell-shaped diffusion curve are no longer suitable to model the upgrade sales of the new generation. To address this problem, we propose a parsimonious and flexible baseline hazard function to capture the declining upgrade hazard rate of a new product generation in the next subsection.

The Exponential-Decay Proportional Hazard Model

Based on our extensive literature review, most existing diffusion models and time-to-purchase models proposed in the prior literature cannot effectively capture such a declining hazard trend. For instance, the hazard function of the classic Bass model (Bass 1969) is monotonically increasing with time. The parsimonious BOXMOD-I framework proposed by Sawhney and Eliashberg (1996), which is a generalization of *Exponential*, *Erlang-2*, and *Generalized-Gamma* distributions, characterizes a non-decreasing baseline hazard function. Therefore, we propose a parsimonious baseline hazard function, which we name as *Exponential Decay* (*Expo-Decay*) baseline hazard function, as follows:

$$h_0(t) = \gamma * e^{-\alpha t}, \alpha, \gamma > 0 \tag{3-3}$$

and the associated survival function is

$$S_0(t) = e^{\left(\frac{\gamma}{\alpha}\right)*(e^{-\alpha t}-1)} \tag{3-4}$$

We refer to γ in the Expo-Decay function as the scale parameter, and α as the decay rate. Although theoretically the Expo-Decay function can capture flat, increasing, or decreasing hazard rates, we are only interested in the declining curve it provides, hence we have $\alpha, \gamma >0$. For expositional convenience, we refer to the Proportional Hazard (PH) Model with Expo-Decay Baseline Hazard Function as the Expo-Decay model.

In the Expo-Decay model, the hazard rate is decreasing overtime, meaning given a potential customer has not upgraded, the probability she will purchase the newly released generation is diminishing overtime. Compared with alternative baseline hazard functions, the Expo-Decay function has two important advantages. First, the Expo-Decay function requires only two parameters, and has as parsimonious a form as the widely applied Weibull function, and simpler than the Expo-Power specification. Second, the Expo-Decay has better

interpretability than other specifications. Specifically, the scale parameter (γ) helps capture the magnitude of incentives to upgrade right after the release of the new generation and the decay rate parameter (α) reflects how quickly the hazard rate declines over time. In contrast, the alternative models such as Weibull, Erlang-2, and Expo-Power are all difficult to interpret.

According to the definition, $h(t, X_i)$ measures the proportional hazard rate during an indefinitely small time interval $(t, t + \Delta t)$. Sometimes, although the underlying survival process is truly continuous, the observable data are grouped into discrete time intervals. In this case, $h(t, X_i)$ cannot stand for the hazard rate for a discrete time interval and fitting the continuous model to grouped survival data might lead to biased estimations. Therefore, based on the continuous survival process, we define a discrete upgrade hazard rate for customer i during time interval j, $(t_{j-1}, t_j]$:

$$\lambda(j, X_i) = Prob\{T_i \leq t_j | T_i > t_{j-1}\} = 1 - e^{-[H(t_j, X_i) - H(t_{j-1}, X_i)]} \quad (3\text{-}5)$$

where $H(t_j, X_i)$ is the cumulative hazard function, $H(t_j, X_i) = \int_0^t h(u, X_i)\, du$. The discrete hazard rate $\lambda(j, X_i)$ represents the probability customer *i* will upgrade during the *j*th interval given she has not upgraded till t_{j-1}.

In general, when customer *i* does not upgrade in the observation window (right censored), the contribution to the likelihood is the probability of survival till the end of the observation window: $L_i = S(t_j, X_i)$. In another scenario when a customer upgrades during the *j*th interval, the likelihood is the probability customer *i* has survived till the end of interval *j*-1 multiplied by the upgrade hazard rate during $(t_{j-1}, t_j]$: $L_i = S(t_{j-1}, X_i) * \lambda(j, X_i)$. Therefore, the likelihood function for customers in a data sample is

$$L = \prod_{i=1}^{N} L_i = \prod_{i=1}^{N} \left[S\left(t_{j-1}, X_i\right) * \lambda(j, X_i)\right]^{\delta_i} \left[S\left(t_j, X_i\right)\right]^{1-\delta_i} \quad (3\text{-}6)$$

where δ_i is a binary indicator of the upgrade status of customer *i*. From Eq. (3-6), we can use the Maximum Likelihood Estimation (MLE) method to obtain the model parameters.

Since the present study examines the upgrade decisions of existing consumers in a multigeneration setting influenced by pre-release virtual adoptions, we only compare the Expo-Decay model with ones that can capture a declining hazard trend, i.e. Weibull and Expo-Power specifications. Details about the empirical analysis are provided in the Empirical Estimation section.

Data Overview

We apply the Expo-Decay model to study consumers' upgrade behaviors for a major sport video game series produced in North America, which is mainly played on gaming consoles such as PlayStation and Xbox. The publisher of the game releases a new generation in the same month every year. For simplicity, the game generation is labeled based on its release year. For instance, the generation released in 2011 is labeled as G-11.

Data Sample

The dataset we use are transactional records of product activations, game playing sessions, and in-game purchases by game console players, which enables us to investigate customers' product adoption, usage, and upgrading behaviors from multiple perspectives.

Players' activations are recorded for generations G-10 through G-16. The video game can be purchased from a brick-and-mortar store or online through the game console. For each generation, ten thousand unique players are sampled based on activation records, resulting in more than 60,000 unique players being tracked across multiple generations of the game series. However, the game playing session records are only available for G-12 through G-16,

and in-game enhancement purchases are collected for G-11 through G-16. To examine the impacts of previous experience (i.e. adoption and usage) on players' future upgrades, we can only utilize playing session records and in-game purchases data starting from G-12 to explain upgrades starting from G-13. In the empirical analysis, we focus on upgrade purchases of G-15, therefore, a sample of 34,584 unique players who have active usage and activation records before the release of G-15 are selected.

Variable Descriptions/Measures

To understand how customers' experience impacts their upgrade decisions, we summarize customers' previous adoption and usage experiences by extracting related covariates from transaction records.

Since early adopters of previous generations tend to upgrade earlier (Huh and Kim 2008), we use WaitDays to denote after the release how long a customer waited to activate the game generation she is currently using. Loyal customers are usually more willing to make future purchases, so we count the number of product generations (NumGens) a customer has previously adopted. The dummy variable Switch indicates whether a customer is a potential switcher or a potential leapfrogger— those who have adopted earlier generations but not the most recent one.

Players' game usage experiences are summarized following the rate of use and variety of use perspectives. Specifically, the number of game sessions a player has played (NumSess) is counted to measure the rate of use. Two variables are generated to denote the variety of use: NumModes, the number of game modes a customer has played, and GiniIndex, the Gini coefficient of the allocation of time among different game modes. Players with high GiniIndex values spend most of the gaming time on a limited number of game modes. The GiniIndex approaches 0 when the player allocates time evenly across all

game modes. Other usage related variables, such as *EnhancePurchase* and *RecentActDay*, are also included. *EnhancePurchase* counts how many in-game enhancement purchases a player has made in one game generation. Since virtual packs are available to enhance players' gaming experience and enhancement packs purchased in one generation cannot be applied in other game generations, these enhancement purchases may imply sunk costs and switching costs and hence impact players' upgrade decisions. *RecentActDay* is defined as the time interval between a customer's latest game session date and the release date of a new generation. Recently active players are expected to have a fresh memory about the game features and show higher willingness to upgrade.

In general, existing players demonstrating active usage patterns are expected to be more open to future technologies. In other words, a customer who has started a larger number of game sessions, played more game modes, made more enhancement purchases, and played the game more recently is expected to demonstrate a higher probability of upgrading.

To rule out the multicollinearity concerns, the variance inflation factor (VIF) analysis is conducted for these intrinsic experience-based variables. Results show all VIF values are below 10, and the high value of *NumModes* reflects the high correlation with *GiniIndex*[1]. Therefore, we remove *NumModes* from further analysis, then all VIF values fall below 3.

[1] The correlation between *NumModes* and *GiniIndex* is -0.81. One possible explanation is that the more modes a customer played, the more time she has to evenly allocate on different modes, leading to a smaller *GiniIndex*.

Table 3.2 Measurements and Descriptions of Explanatory Variables

Variable	Label	Mean	SD	Min	Max
Past Adoption Experience					
Number of generations activated by the customer	*NumGens*	1.51	0.40	1	3
Time interval between release dates and activation dates for adopted generations	*WaitDays*	167.5	191.38	-11	1096
Whether the customer has activated the most recent generation	*Switch*	0.51	0.50	0	1
Past Usage Experience					
The number of game sessions the customer has played	*NumSess*	36.9	68.60	1	1920
The number of game modes the customer has played	*NumModes*	4.49	3.47	1	25
The Gini coefficient of the allocation of time spent on different game modes	*GiniIndex*	0.90	0.06	0.56	0.97
The number of in-games enhancement purchases a customer has made	*EnhancePurchase*	2.12	19.47	0	1494
Time interval between a customer's latest game session and the release date	*RecentActDay*	304.1	282.56	0	1019

Model-Free Evidence

In the multigeneration video game series, the proportion of existing customers in the composition of adopters of the new product generation is increasing during the data period (Figure 3.2). The abundant transactional records of existing customers enable the examination of the impact of consumers' experience on future upgrade decisions.

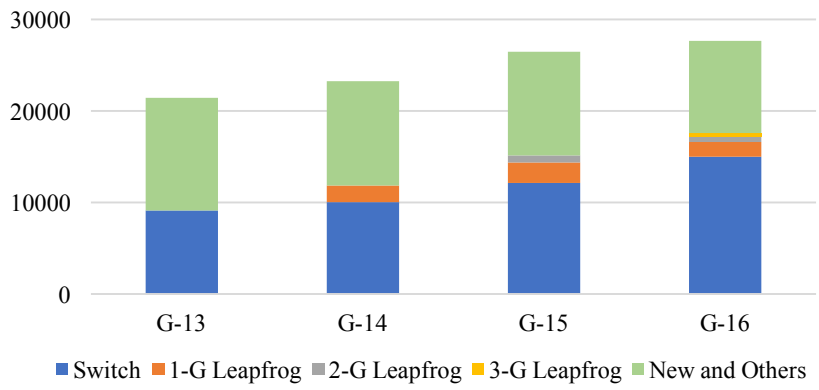
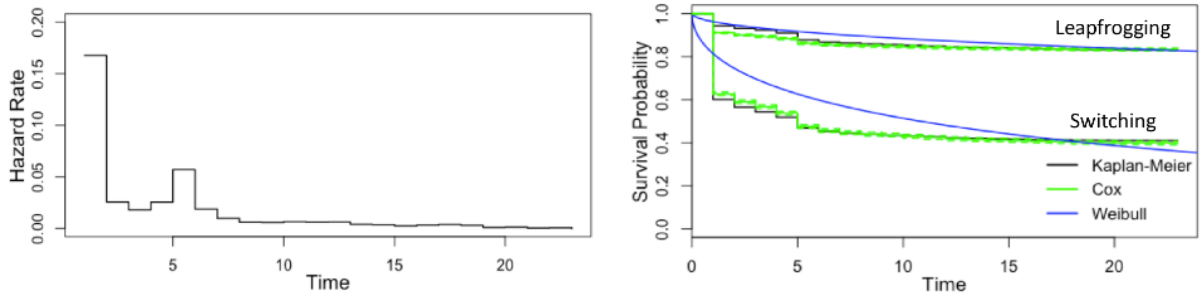


Figure 3.2 Composition of Sales for Each Game Generation

In Figure 3.2, n-G leapfrog denotes a player has skipped n generations in the middle and activates the focal generation. Due to data truncation problem, the new and others adopters include new adopters and leapfroggers who have skipped more generations than we could track. However, the 1-G and 2-G leapfrogs can be identified but only account for a small proportion (around 7% to 10%) in sales. It is also worth noting that among all existing customers, the switching players is taking a large proportion in adoption.



(a) (b)

Figure 3.3 Kaplan-Meier Estimation of Hazard Rate and Survival Function

Before specifying any baseline hazard function, we apply the Kaplan-Meier method to estimate the upgrade hazard rate (Figure 3.3-a). Model-free estimations show that in the first month after release and, to a less extent, the holiday season, existing customers are more

likely to upgrade. The upgrade hazard rate decreases over time and drops close to 0 after 12 months (Figure 3.3-a), after which a newer generation is released.

On average, it takes existing consumers around 3.9 months before upgrading to a new generation. For switching customers, it takes them 2.46 months to upgrade, while for leapfrogging customers, upgrades can take 4.92 months on average. Potential switchers and potential leapfroggers are expected to demonstrate asymmetric upgrade decisions (Jiang and Jain, 2012). Due to their knowledge with the more recent product generation, potential switchers perceive the complexity of a new product generation as being lower and the compatibility for a new generation as being higher considering the incremental technology improvements. Figure 3.3-b shows potential switching customers are expected to demonstrate a relative low survival probability, in other words, a higher probability to upgrade. The heterogeneity between switching and leapfrogging customers is modeled through the switch dummy variable.

Empirical Estimation

In this section, we use the game adoption and usage data described in the previous section to (i) examine the impacts of customers' experience on the time to upgrade to a new game generation, and (ii) evaluate the performance of the Expo-Decay model in relation to alternative models.

Impact of Adoption and Usage Experience on Time-To-Upgrade

We estimate the proportional hazard models with different specifications. Dummy variables, corresponding to the first month after release and the holiday month respectively, are included to capture the abnormalities. Estimation results are summarized in Table 3.3. It is evident that all intrinsic experience-based variables have significant impacts on existing players' time to upgrade decisions.

Table 3.3 Proportional Hazard Model Estimation Results

	Cox	**Weibull**	**Expo-Power**	**Expo-Decay**
Covariates	Coefficient (Std.)	Coefficient (Std.)	Coefficient (Std.)	Coefficient (Std.)
NumGens	0.4352 (0.0142) ***	0.4565 (0.0141) ***	0.4483 (0.0143) ***	0.4479 (0.0143) ***
WaitDays	-0.0016 (0.0001) ***	-0.0015 (0.0001) ***	-0.0015 (0.0001) ***	-0.0015 (0.0001) ***
Switch	0.3066 (0.0328) ***	0.3379 (0.0335) ***	0.3330 (0.0143) ***	0.3329 (0.0333) ***
NumSess	0.0016 (0.0001) ***	0.0026 (0.0001) ***	0.0025 (0.0001) ***	0.0025 (0.0001) ***
GiniIndex	-1.4165 (0.1738) ***	-0.9374 (0.1373) ***	-1.3642 (0.1895) ***	-1.3632 (0.1740) ***
EnhancePurchase	0.0012 (0.0003) ***	0.0016 (0.0003) ***	0.0017 (0.0003) ***	0.0017 (0.0003) ***
RecentActDay	-0.0028 (0.0001) ***	-0.0026 (0.0001) ***	-0.0027 (0.0001) ***	-0.0027 (0.0001) ***
α	––	0.1119 (0.0116) ***	1.0744 (0.0317) ***	0.1687 (0.0039) ***
γ	––	0.7649 (0.2841) **	0.1077 (0.0206) ***	0.1292 (0.0211) ***
θ	––	––	-0.1443 (0.0102) ***	––
First Month Dummy	––	-0.5406 (0.0611) ***	0.5492 (0.0960) ***	0.5305 (0.0856) ***
Holiday Dummy	––	0.1124 (0.0143) ***	0.1505 (0.0269) ***	0.1525 (0.0254) ***
BIC	186796.9	56672.71	56320.72	56315.81

Although the absolute values of estimated coefficients vary due to different model specifications, the signs of coefficients and significance levels are consistent. Empirical results show that rate of use (i.e. *NumSess*) has a positive impact on customers' upgrade

probability. However, the negative influence from variety of use (i.e. *GiniIndex*) is counter-intuitive, reflecting that specialized players, who spend most of the gaming time on only a few game modes (low usage variety and large *GiniIndex*), are less likely to upgrade to a new generation. One explanation is that specialized players, after exploring different game modes, might finally find the game mode(s) they like the most and spend most of the time enjoying these game modes. Hence a high Gini-Index indicates the player is satisfied with features of the current game generation in use, as a result, the time to upgrade will be longer. This is analogous to the current trend in the smartphone industry — customers' satisfaction with old-generation smartphones postpones upgrades (Martin and FitzGerald, 2018).

Investments in enhancement packs in previous game generations do not delay players' time to upgrade. On the contrary, customers who have made more enhancement purchases in previous generations are more likely to upgrade. Although a relatively small proportion (around 18%) of players have ever made in-game purchases, they represent high-end consumers with a higher willingness to upgrade. In addition, customers who are active more recently (a smaller *RecentActDay*) demonstrate higher upgrade probabilities.

Customers' previous adoptions are found to have significant influences on the future upgrade decisions as well. If a customer has adopted one more generation in the game series, the hazard rate to upgrade will increase by nearly 50%. In particular, potential switching customers are more likely to upgrade compared to potential leapfrogging customers. The negative influence of *WaitDays* is consistent with the literature — early adopters tend to upgrade earlier.

It is worth noting that when α in the Expo-Power function approaches 1, it nests to the Expo-Decay function, and empirical evidences show the estimated Expo-Power model

reduces to an Expo-Decay model. Therefore, the Expo-Decay model with both first-month and holiday dummies is a reasonable specification to estimate time-to-upgrade decisions in a multigeneration product series in the presence substantial pre-release virtual adoptions.

Model Comparisons

For further evaluation, the proposed Expo-Decay models are compared against benchmark methods in predicting upgrade sales at the aggregate level. The predicted number of monthly upgrade sales is the aggregation of individual upgrade probabilities over time (Gupta, 1991). Since the Cox PH model does not specify a baseline hazard function, the comparison is between the prediction results by the Expo-Decay model and those of parametric benchmark methods, i.e. Weibull, and Expo-Power models.

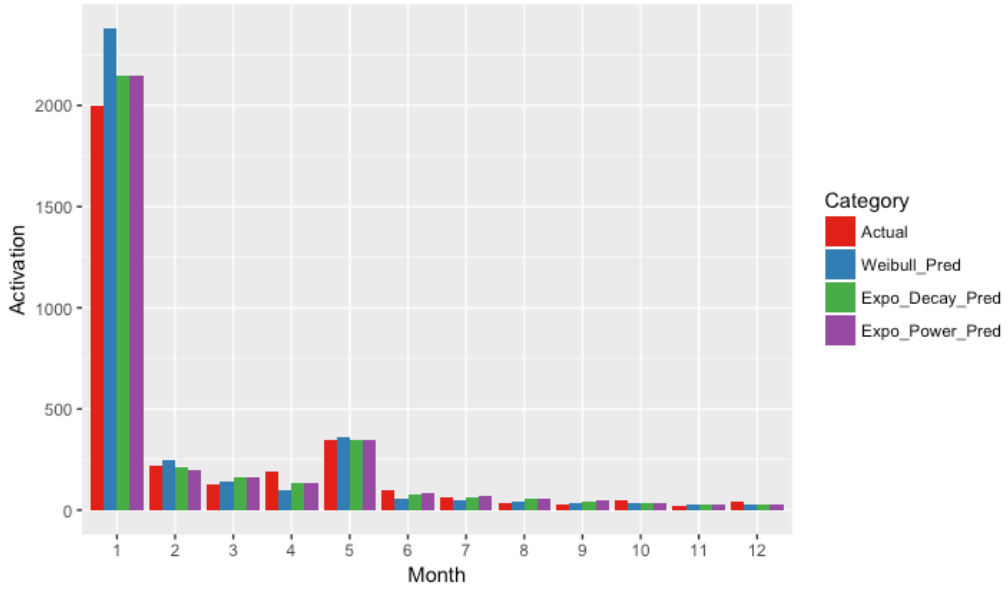


Figure 3.4 Predicted Versus Actual Monthly Upgrade Sales

For validation, we use 75% of unique players' records to train the model, and the remaining 25% for out-of-sample validation. By summing up the individuals' probability to

upgrade at each discrete time interval, the predicted monthly upgrade sales[2] and the actual upgrade sales are depicted in Figure 3.4, which shows the Expo-Decay model can forecast the upgrade sales as well as the flexible Expo-Power model.

To compare the prediction performances in a more systematic manner, we use four metrics, including Theil's inequality coefficient, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The Theil's inequality coefficient (Gupta 1991) is defined as:

$$U = \frac{\sqrt{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2/T}}{\sqrt{\sum_{t=1}^{T}(y_t)^2/T} + \sqrt{\sum_{t=1}^{T}(\hat{y}_t)^2/T}} \tag{3-7}$$

in which y_t and $\hat{y}_t$ are actual and predicted number of upgrades in month *t*. The coefficient *U* ranges from 0 to 1, where a smaller value means a better prediction performance.

Table 3.4 Comparison of Upgrade Sales Predictions

	Weibull	**Expo-Power**	**Expo-Decay**
Theil's Coefficient	0.0269	0.0198	0.0198
MSE	559.55	302.41	303.36
MAE	15.36	11.95	11.27
MAPE	74.74%	36.36%	35.24%

[2] Approximated by the number of activations.

From the comparison in Table 3.4, we conclude that the Expo-Decay model performs as good as the Expo-Power model, and significantly better than the Weibull models. Coupled with the fact that the Expo-Power model has more parameters to be estimated and is far more difficult to interpret, we conclude that the proposed Expo-Decay model is a superior model in predicting time-to-upgrade decisions by customers in the presence of successive product generations.

With all empirical results considered, we conclude that the Expo-Decay model is an effective method in explaining and predicting existing customers' time-to-upgrade decisions in a multigeneration setting when pre-release virtual adoptions account for the majority in upgrade sales.

Model Extensions

The baseline Expo-Decay model depicts the underling decaying trend of existing users' upgrade intention. However, factors that are not observable to researchers and firms might also influence existing users' upgrade decisions. Meanwhile, in the baseline model, users' adoption and usage history is summarized into aggregated variables, which leaves possibility for modeling a customer's entire adoption history. In this section, we develop extended models and compare with the baseline Expo-Decay model in estimations and upgrade predictions.

Frailty Expo-Decay Model

In reality, only a proportion of users' profile or shopping and usage history is observable to researchers or the company. Due to privacy concerns, sensitive information regarding customers' identity is not accessible. These unobservable factors may still impose significant impacts on existing users' upgrade hazards.

Table 3.5 Estimation Results for Extended Models

	Expo-Decay	**Frailty Expo-Decay**	**Expo-Decay-II**	**Frailty Expo-Decay-II**	**Time-Variant Expo-Decay**
Variables	Coefficient (Std.)	Coefficient (Std.)	Coefficient (Std.)	Coefficient (Std.)	Coefficient (Std.)
NumGens	0.4479 (0.0143) ***	0.6538 (0.0260) ***	0.6729 (0.0150) ***	0.9154 (0.0284) ***	0.4431 (0.0143) ***
WaitDays	-0.0015 (0.0001) ***	-0.0015 (0.0001) ***	––	––	-0.0016 (0.0001) ***
Switch	0.3330 (0.0334) ***	0.5006 (0.0460) ***	––	––	0.3182 (0.0312) ***
NumSess	0.0025 (0.0001) ***	0.0059 (0.0004) ***	0.0032 (0.0001) ***	0.0071 (0.0004) ***	0.0013 (0.0001) ***
GiniIndex	-1.3626 (0.1783) ***	-1.8418 (0.2395) ***	-1.7273 (0.2006) ***	-1.8905 (0.2513) ***	-1.650 (0.1705) ***
EnhancePurchase	0.0017 (0.0003) ***	0.0107 (0.0019) ***	0.0020 (0.0003) ***	0.0116 (0.0019) ***	0.0015 (0.0003) ***
RecentActDay	-0.0027 (0.0001) ***	-0.0028 (0.0001) ***	-0.0028 (0.0001) ***	-0.0031 (0.0001) ***	-0.0026 (0.0001) ***
α	0.1687 (0.0039) ***	0.1476 (0.0042) ***	0.2069 (0.0084) ***	0.1608 (0.0064) ***	0.1254 (0.0042) ***
γ	0.1291 (0.0214) ***	0.1781 (0.0413) ***	0.1064 (0.0201) ***	0.1132 (0.0267) ***	0.1853 (0.0290) ***
σ^2	––	0.9941 (0.0844) ***	––	0.9619 (0.0817) ***	––
α_{G-3}	––	––	0.0029 (0.0011) **	0.0018 (0.0010) •	––
α_{G-2}	––	––	0.0040 (0.0015) **	0.0017 (0.0014)	––
α_{G-1}	––	––	0.0391 (0.0064) ***	0.0539 (0.0111) ***	––
ϕ_{G-3}	––	––	0.0271 (0.0088) **	0.0208 (0.0130)	––
ϕ_{G-2}	––	––	0.0479 (0.0114) ***	0.0421 (0.0224) •	––
ϕ_{G-1}	––	––	0.1415 (0.0102) ***	0.1550 (0.0134) ***	––
BIC	56315.81	**56059.33**	56945.27	56683.52	56365.58

As a result, we introduce a random variable θ into the baseline hazard function to represent the unobserved customer heterogeneity: $h(t|X_i,\ \theta) = \theta * h_0(t) * e^{X_i\beta}$.

Without loss of generalizability, we assume θ follows a gamma distribution with an expected value of 1, $\theta \sim Gamma(\frac{1}{\sigma^2}, \frac{1}{\sigma^2})$, $E(\theta) = 1$ and $Var(\theta) = \sigma^2$. Correspondingly, the survival function is: $S(t|X_i,\ \theta) = \exp\{-\int_0^t \theta * h_0(\tau) * e^{X_i\beta}\, d\tau\}$.

By introducing the frailty Expo-Decay model, the unobserved heterogeneity of existing users is modeled. From the empirical estimation results in Table 3.5, the Frailty Expo-Decay model provides the best model fitting by introducing just one more parameter (in term of BIC). The estimated variance for the unobservable parameter θ is around 0.9941. However, the Frailty Expo-Decay model does not perform as well as the baseline Expo-Decay model in forecasting aggregate upgrade sales (Table 3.6) and predicting individual upgrade decisions (Figure 3.4). Without unobservable factors, users' previous adoption and usage experience can work as powerful indicators of their future upgrade decisions. Introducing the unobservable heterogeneity factor leads to overfitting when predicting upgrade purchases.

Expo-Decay II Model

In the baseline Expo-Decay model, an existing user's previous adoptions is aggregated into cumulative variables (e.g. *NumGens* and *WaitDays*) reflecting the user's experience with the product line. From another perspective, the user's experience of adopting previous product generations would cast discrete impact on the user's future decisions, which will have a long-lasting effect. Instead of using aggregated variables to denote a user's adoption history, we model the impact of discrete adoption behaviors into the hazard function and presume the influence of these events will decay over time.

To model the influence of adoptions of previous product generations, we extend the Expo-Decay model by integrating the exciting point process method (Xu et al. 2014), which assumes influences of previous adoptions events are additive to the hazard rate:

$$h(t, X_i) = h_0(t)e^{X_i'\beta} + \sum_{g=1}^{G-1} 1_{i,g} * \alpha_g * e^{-\phi_g * (t-\tau_{i,g})} \quad (3\text{-}8)$$

where $1_{i,g}$ indicates whether customer *i* has adopted generation *g*, α_g measures the influence of the adoption of generation *g* with a decaying factor ϕ_g, and $\tau_{i,g}$ denotes when customer *i* activated generation *g*. It is worth noting that the influence of previous adoptions also decays exponentially over time, which is the reason the extended model in Eq. (3-8) is named as *Expo-Decay II* model.

Based on the empirical estimation results from Table 3.5, in the Expo-Decay-II model, adoptions of previous game generations show different impact on future upgrade: the activation of more recent game generation has a larger impact but the effects decays faster. However, the Expo-Decay II model does not improve on modeling fitting. Although modeling the previous adoption events minimizes errors in model fitting, the BIC measure penalizes the newly introduced parameters. Compared to the baseline Expo-Decay model, the aggregated users' adoption features can better explain their upgrade behavior without modeling each action discretely. In prediction, it is inferior to the baseline model as well (Table 3.6 and Figure 3.4).

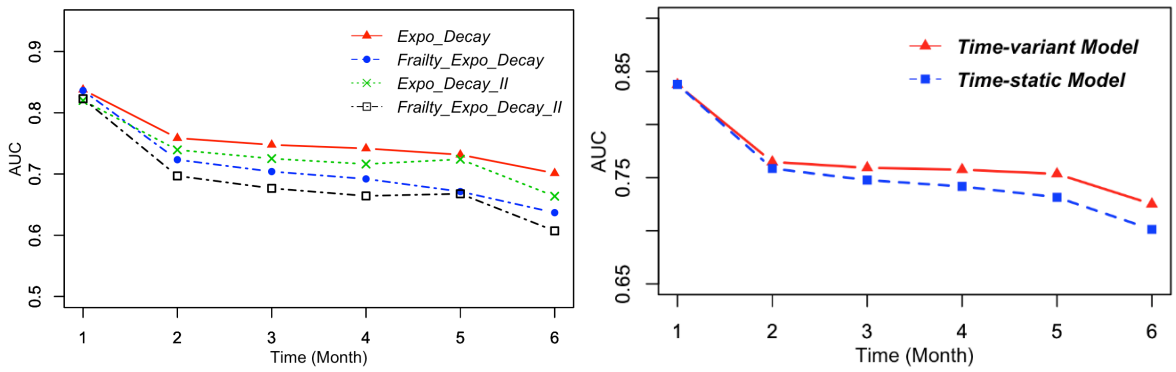
In another extension, we combine the frailty model with the Expo-Decay II model by modeling the unobserved heterogeneity and discrete adoptions into the baseline model at the same time. Empirical estimations show that after considering consumer heterogeneity, only the adoption experience of the most recent generation (G-1) significantly impacts upgrading to the next generation (α_{G-1} and ϕ_{G-1}). However, the Frailty Expo-Decay-II model

performs the worst among the baseline model and extensions, mainly due to overfitting (Table 3.6 and Figure 3.4).

Table 3.6 Aggregate Upgrade Sales Prediction

PHMs	Expo-Decay	Frailty Expo-Decay	Expo-Decay-II	Frailty Expo-Decay-II
Theil's Measure	0.0198	0.0210	0.0207	0.0223
MSE	303.36	334.45	325.5	376.23
MAE	11.27	12	11.68	13.05
MAPE	35.24%	36.63%	37.30%	43.24%

Although the abovementioned model extensions do not lead to any improvements in predictions, these extended models can be readily applied for future research or practice. In different context, the relative performances may vary.



(a) (b)

Figure 3.5 AUC of Individual Upgrade Predictions by Extended Models

The Time-Variant Model Extension

The baseline Expo-Decay model summarizes and utilizes existing users' information at the release date. Based on the cumulated experience, the user adoption and usage based variables are applied to explain and predict their future upgrade decisions. However, users' experience is not static, which would be a limitation when the prediction time window is not short. If the model could not effectively capture the user's cumulated experience evolving overtime, it would fail to accurately understand the decision-making process or evaluate the prediction power of each indicator.

In the baseline model, the user's behavior after the release date is not modeled, but events happening later may strongly indicate the postpone of an upgrade purchase. For instance, one user adopted G-13 and G-14 after 1.4 months and 4.9 months respectively. After G-15 is released, the estimated upgrade timing by the baseline model is around 6 months. However, the user adopts G-12 and play G-12 for a few sessions. Such backward adoption behavior delays her upgrade purchase of G-15, for which she adopts in one year after its release date.

In this subsection, we propose a time-variant extension to the Expo-Decay model, in which the user's behavior related variables are time-variant: $h\big(t, X_i(t)\big) = h_0(t) \cdot \exp^{[X_i(t)'\beta]}$.

The experience-based variables are summarized at the beginning of each discretized time interval and then applied to model the hazard rate for the next coming interval. While the variables are changing overtime, the coefficient β is fixed.

Providing more up-to-date information will improve the performance in prediction. Since the time-variant model requires variables to be updated overtime, it is not appropriate

to be applied in forecasting upgrade sales at the release date. Thus, we evaluate the performance of time-variant model on predicting individual user's upgrade decisions using time-dependent ROCs.

Conclusions

Continuous quality improvements and frequent releases of new generations in a product series is a common practice by businesses, which helps them counter competition, generate upgrade purchases, and maintain market share. In the presence of successive product generations, it is important to understand customers' upgrade decisions when a new product generation becomes available. In particular, we are interested in a scenario where pre-release virtual adoptions account for the majority of upgrade sales of a new product generation, and the upgrade hazard rate exhibits a declining pattern after the product release. Given there is no good model options exist in the prior literature, this study proposes a survival model, specifically a proportional hazard model, with an Exponential-Decaying baseline hazard function (Expo-Decay model) to examine how existing customers' experience (i.e. adoption and usage behavior) impacts their upgrade decisions.

This study makes an important methodological contribution to the existing survival analysis literature. Specifically, the Expo-Decay model we propose can help explain the declining upgrade hazard rate of a new product generation when pre-release virtual adoptions account for the majority of upgrade sales. The Expo-Decay model is parsimonious, easy to interpret, and delivers superb model fit and prediction performance when compared to existing parametric proportional hazard models, hence it has the potential of wide application in future academic research. Furthermore, the extended Expo-Decay-II model provides an innovate way to capture the influence of discrete previous adoption events by integrating a point process. In addition, the Frailty Expo-Decay model can effectively explain the

unobservable customer heterogeneity and the Time-Variant Expo-Decay model includes time-variant features which could describe the evolving trajectory of customers' adoption and usage behavior. Although the performances of these extended models vary in model fitting and upgrade predictions (at aggregate and individual levels), these model extensions provide ready-to-use specifications for future research and practice applications.

This study also contributes to our understanding regarding the factors that help predict customers' time to product upgrade. Although the existing literature have identified some driving factors that may influence users' upgrade decisions, this study fills a gap by linking customers' previous adoption and usage experience to future upgrade purchases. Using a rich dataset for a video game series, we find that consumers' prior adoption and usage experience has a significant impact on their likelihood of upgrade and time to upgrade purchase. In particular, we find that (i) after a new product generation is released, potential switching customers who are using the latest available generation are more willing to upgrade; (ii) heavy users of the product series tend to upgrade earlier; and (iii) specialized customers (those focusing on a relatively small number of product functions) demonstrate a higher upgrade probability.

The survival model proposed and the empirical findings also have important managerial implications. The Expo-Decay model can be used to predict future upgrade sales, which can help a firm better manage the production, promotion, and distributions of a new product generation. The findings regarding how customers' prior adoption and usage experience affects their time-to-upgrade can help the firm segment the market, design and deliver more specialized products for different types of customers, and develop personalized

promotions to target customers. Such tailored marketing efforts can improve customer satisfaction and the efficiency of operations, leading to better and longer-term profitability.

This current study is not without limitations, which leaves several interesting future research directions. First, we validate the proposed Expo-Decay model using video games dataset only. A future study could test the model for other product categories and possibly develop a more specialized model based on the observed sales growth patterns. Second, we do not consider the impact of marketing mix variables such as price and promotion on customers' time-to-upgrade decisions. It could be interesting to extend the proposed Expo-Decay model to capture the impact of marketing mix variables. Third, one could check the various information channels (e.g., social media) through which customers can collect information about a new product generation, and examine whether different information channels affect customers' upgrade decisions differently.

References

Alba, J.W. and J.W. Hutchinson (1987). "Dimensions of consumer expertise," *Journal of Consumer Research*, 13(4): 411-454.

Albuquerque, P., and Y. Nevskaya (2012). "The impact of innovation on product usage: A dynamic model with progression in content consumption".

Bardhan, I., Oh, J.H., Z. Zheng and K. Kirksey (2014). "Predictive analytics for readmission of patients with congestive heart failure," *Information Systems Research*, 26(1): 19-39.

Bayus, B. L. and S. Gupta (1992). "An empirical analysis of consumer durable replacement intentions," *International Journal of Research in Marketing*, 9(3): 257-267.

Chang, M.K., W. Cheung, and V.S. Lai (2005). "Literature derived reference models for the adoption of online shopping," *Information & Management*, 42(4): 543-559.

Christensen, C.M. (1992). "Exploring the limits of the technology S-curve. Part I: component technologies," *Production and Operations Management*, 1(4): 334-357.

Cox, D.R. (1972). "Regression models and life-tables," *Journal of the Royal Statistical Society B*, 34 (2): 187-220.

Davis Jr, F. D. (1986). "A technology acceptance model for empirically testing new end-user information systems: Theory and results," Do*ctoral Dissertation, Massachusetts Institute of Technology*.

Dee Dickerson, M. and J.W. Gentry (1983). "Characteristics of adopters and non-adopters of home computers," *Journal of Consumer research*, 10(2): 225-235.

Edwards, J. (2016). "iPhone 7 is poised for record sales — here is the stat that proves it," Business Insider. Retrieved from http://www.businessinsider.com/iphone-7-will-see-record-sales-due-to-existing-customer-upgrades-installed-user-base-2016-5?r=UK&IR=T.

Ellen, P.S., W.O. Bearden, and S. Sharma (1991). "Resistance to technological innovations: an examination of the role of self-efficacy and performance satisfaction," *Journal of the Academy of Marketing Science*, 19(4): 297-307.

Entertainment Software Association (2016). "Essential facts about the computer and the video game industry." Retrieved from http://www.theesa.com/wp-content/uploads/2016/04/Essential-Facts-2016.pdf.

Gelper, S., R. Peres, and J. Eliashberg (2014). "Pre-release word-of-mouth dynamics: The role of spikes," *Erasmus University Rotterdam working paper*.

Golder, P. N., and G. J. Tellis (1997). "Will it ever fly? Modeling the takeoff of really new consumer durables," *Marketing Science*, 16(3): 256-270.

Grewal, R., R. Mehta, and F. R. Kardes (2004). "The timing of repeat purchases of consumer durable goods: The role of functional bases of consumer attitudes," *Journal of Marketing Research*, 41(1): 101-115.

Gerlach, J., R.M. Stock, and P. Buxmann (2014). "Never Forget Where You're Coming from: The Role of Existing Products in Adoptions of Substituting Technologies," *Journal of Product Innovation Management*, 31(S1):133-145.

Gupta, S. (1991). "Stochastic models of interpurchase time with time-dependent covariates," *Journal of Marketing Research*: 1-15.

Huh, Y. E., and S. H. Kim (2008). "Do early adopters upgrade early? Role of post-adoption behavior in the purchase of next-generation products," *Journal of Business Research*, 61(1): 40-46.

Jiang, Z. and D.C. Jain (2012). "A generalized Norton–Bass model for multigeneration diffusion," *Management Science*, 58(10): 1887-1897.

Kameda T. and J.H. Davis (199). "The function of the reference point in individual and group risk decision making," *Organizational Behavior and Human Decision Processes*, 46(1): 55-76.

Kim, N., Srivastava, R.K. and Han, J.K., 2001. Consumer decision-making in a multi-generational choice set context. *Journal of Business Research*, 53(3), pp.123-136.

Kim, S.H. and V. Srinivasan (2009). "A Conjoint-Hazard Model of the Timing of Buyers' Upgrading to Improved Versions of High-Technology Products," *Journal of Product Innovation Management*, 26(3): 278-290.

Martin, T. W. and D. FitzGerald (2018). "Your Love of Your Old Smartphone Is a Problem for Apple and Samsung," The Wall Street Journal. Retrieved from https://www.wsj.com/articles/your-love-of-your-old-smartphone-is-a-problem-for-apple-and-samsung-1519822801.

Okada, E.M. (2001). "Trade-ins, mental accounting, and product replacement decisions," *Journal of Consumer Research*, 27(4): 433-446.

Okada, E.M. (2006). "Upgrades and new purchases," *Journal of Marketing*, 70(4): 92-102.

Purohit, D. (1995). "Playing the role of buyer and seller: The mental accounting of trade-ins," *Marketing Letters*, 6(2): 101-110.

Rijnsoever, F.J. and H. Oppewal (2012). "Predicting early adoption of successive video player generations," *Technological Forecasting and Social Change*, 79(3): 558-569.

Rogers, E.M. (2003). "Elements of diffusion," *Diffusion of Innovations*, 5(1.38).

Sääksjärvi, M. and M. Lampinen (2005). "Consumer perceived risk in successive product generations," *European Journal of Innovation Management*, 8(2): 145-156.

Seetharaman, P.B. and P.K. Chintagunta (2003). "The proportional hazard model for purchase timing: A comparison of alternative specifications," *Journal of Business & Economic Statistics*, 21(3): 368-382.

Shih, C. F., and A. Venkatesh (2004). "Beyond adoption: Development and application of a use-diffusion model," *Journal of Marketing*, 68(1): 59-72.

Therneau, T., C. Crowson and E. Atkinson (2017). "Using time dependent covariates and time dependent coefficients in the cox model," Survival Vignettes.

Thong, J.Y., S.J. Hong and K.Y. Tam (2006). "The effects of post-adoption beliefs on the expectation-confirmation model for information technology continuance," *International Journal of Human-Computer Studies*, 64(9): 799-810.

Tseng, F.M. and H.Y. Lo (2011). "Antecedents of consumers' intentions to upgrade their mobile phones," *Telecommunications Policy*, 35(1): 74-86.

Van Nes, N. and J. Cramer (2008). "Conceptual model on replacement behavior," *International Journal of Product Development*, 6(3-4): 291-309.

Zhu, R., X. Chen, and S. Dasgupta (2008). "Can trade-ins hurt you? Exploring the effect of a trade-in on consumers' willingness to pay for a new product," *Journal of Marketing Research*. 45(2): 159-170.

CHAPTER 4. OPTIMAL MAINTENANCE POLICY FOR CONSOLIDATED DATA REPOSITORY UNDER INFINITE TIME HORIZON

Modified from a manuscript to be submitted to Production and Operations Management

Xinxue Qu and Zhengrui Jiang

Ivy College of Business, Iowa State University, Ames, IA, USA

Abstract

With the development of various information technologies and wide implementation of enterprise information systems, data is being generated at a speed never seen before. Moreover, the large volume data sets can be collected from various data sources (e.g. transactional systems, social media) in different formats (e.g. text, picture, geo-graphical records). On the other hand, to support Business Intelligence applications and decision-makings, organizations have to keep their information asset up-to-date to avoid possible mistakes in daily operations and strategic planning. In the age of big data, the complexity of the information asset maintenance for firms cannot be easily addressed using existing methods. Existing maintenance policies in the literature are either static in nature or difficult to operationalize, which mostly focus on the maintenance in a finite time horizon. Therefore, in this study, we model the information asset maintenance problem as a Markov decision process and extend the time-based dynamic synchronization in an infinite planning horizon. Given the maintenance context, we are able to prove the existence of the optimal control limit at each decision epoch and propose an optimal control policy, which is easy to operationalize and leads to significant cost savings.

Keywords: Markov decision process, information asset maintenance, optimization, infinite horizon.

Introduction

In recent years, with the wide application of Business Intelligence systems and the rapid development of Machine Learning and Deep Learning methods, organizations are relying more on analytics to gain a competitive advantage. According to 2018 MIT Sloan Management Review Global Executive Study and Research Report[1], around 59% of the participated managers agree that their companies are applying or move to deploy analytics tools to gain a competitive advantage in the market. 49% of the respondents in 2017 report that they are able to effectively using data to guide future strategies. In future decision-markings, it would depend more heavily on the firms' data/information assets and analytics tools. Moreover, to train a sophisticated analytics model (e.g. a deep neural network), a large amount of data is necessary, which emphasizes the importance of information assets in companies' strategic shift toward analytics.

In the past few decades, the development of information systems has facilitated organizations with more channels to collect data and accumulate information assets from every possible dimension. Transaction Processing Systems provide data generated from every business transactions, including sales transactions, procurement, customer engagement, product manufacturing, etc. With wireless technologies, more and more sensors would have been deployed in the entire business processes to monitor every single operation and collect the data for further analysis. At the same time, with the prosperity of social media and social platform, various types of user generated content become available to companies. Different from Web 1.0, users are allowed to post any format of information online with Web 2.0 technologies. Usually, user generated contents are unstructured, which challenges traditional

[1] https://sloanreview.mit.edu/projects/using-analytics-to-improve-customer-engagement/

design of transactional database systems and data warehouse systems. A few advanced data collection and storage technologies (e.g. web-crawlers, Distributed File Systems) have been implemented to collect useful information. The collected information would provide a more comprehensive description about the consumers and the business environment, firms can better understand the requirements from the users and customize their recommendation systems to improve customer satisfaction and increase the rate of retention.

Although the abundant data resources are the most valuable assets for organizations' strategic shift to analytics, big data may not always lead to revenue increases. According to Gartner's Data Quality Market Survey[2], poor data quality is also hitting organizations leading to an average $15 million annual financial cost in 2017. What's worse, nearly 60% of respondent organizations do not event pay attention to or just do not have an effective way to measure the annual financial cost of poor quality data. In the age of big data, huge amount data (high *volume*) from various sources in different formats (high *variety*) is generated in real-time (high *velocity*). This fast change digital world leaves the information assets maintenance in organizations outdated more rapidly than ever. If not maintained properly, data scientists and decision-makers may draw their conclusion based on these *stale data*, which would generate biased estimation results and inaccurate market predictions, and indirectly lead to severe business losses. Therefore, it is crucial for organizations to develop efficient and effective information asset maintenance rules to reduce the loss incurred by low-quality information.

In literature, there have been discussions regarding the optimal maintenance policies. Although researchers attempt to achieve real-time data synchronization to incorporate data

[2] https://www.gartner.com/smarterwithgartner/how-to-stop-data-quality-undermining-your-business/

changes from different sources into the centralized information system, synchronization of the central consolidated system cannot be achieved in real time. On one hand, synchronizations take time and the more data changes accumulated, the longer it takes to run. On the other hand, synchronization at real time will compete for computing resources leading to slow-down to regular business operations. Therefore, studies try to find the optimal solution to when and how frequent to synchronize the consolidated information systems. Frequent systems synchronizations would reduce data staleness problem and improve the quality of decision-makings, but incur large synchronization costs. Nevertheless, if not synchronized properly, stale data will cause problems in the business decisions leading to severe business losses. The optimal maintenance policy should strike a balance between these two types of economic cost and schedule an easy-to-operationalize policy for system administration.

Following this direction, studies have proposed different policies, specifically, time-based, update-based, and query-based policies. However, most extant discussions are focusing on a finite planning horizon and the proposed policies are static in nature. In this study, we will extend the time-based dynamic synchronization policy to an infinite planning horizon to optimize the system maintenance.

In the next section, we will review related literature and compare the differences in the proposed maintenance policies. Then, we follow a stochastic process framework to model the maintenance problem as a Markov decision process. Next, we will search for the optimal solution to the problem, and conduct comparison among policies. The paper will be concluded with discussions and future research directions.

Related Literature

The importance of system maintenance and synchronization frequency has been identified as an important practical and research questions (Jarke et al. 2000), and there are a few different research streams related to this study using quantitative analysis methods.

In one research stream, studies focus on proposing optimal policies to reduce data staleness cost. The system synchronization schedule is geared toward maintaining the freshness of information. Xiong and Ramamritham (2004) propose a periodic policy for information system maintenance to achieve data validity. However, this research direction omits the system synchronization cost, which is an important factor impacting the optimal system maintenance schedule.

There are another stream of studies incorporating both system synchronization cost and the information staleness cost into consideration and proposing different policies for the system maintenance. Segev and Fang (1991) develop a stochastic model to compare the time-based and query-based refresh policies. Dey et al. (2006) analytically compare three different periodic policies (i.e. time-based, update-based, and query-based), and show that query-based policy is inferior to time-based policy in terms of total costs and update-based policy is the best among the three. However, the time-based policy is the easiest to operationalize, and cost savings by the update-based policy are not significant. As a result, they propose the time-based policy, which synchronize the system following an optimized time interval, is the best in practice. More recently, Fang et al. (2013) and Zong et al. (2017) propose query-based dynamic policies for constantly changing database systems, which check system states whenever an information query comes to the system and a synchronization will be scheduled if the cumulated number of updates reaches the optimal threshold. In a different context, Dey et al. (2015) propose a hybrid policy for system security patches maintenance. The

hybrid policy is composed with one optimal time interval and one optimal updates threshold. The system synchronization will be triggered either the lapsed time has gone beyond the optimal time interval or the cumulated updates have reached the optimal threshold. Different from the above policies, Qu and Jiang (forthcoming) proposed a time-based dynamic synchronization policy, which schedules system check according to a predetermined time interval and only run the synchronization when the stale data accumulates to certain thresholds. Compared to other policies, the time-based dynamic policy is dynamic in nature and easy to operationalize. Therefore, in this study, we extend the time-based dynamic policy to a more realistic application scenario, an infinite planning horizon, for the system maintenance.

Given the policies discussed in the literature, the system maintenance policy proposed in this study differs from the literature in the following aspects: (i) We discuss the information system maintenance under an infinite planning horizon, while existing studies only concentrate the synchronization schedules under a finite time horizon. (ii) The time-based dynamic synchronization policy we apply can schedule system checks during off business hours, thus synchronizations of the system would avoid any disruption to regular business operations and save the *business disruption cost*. The best-performing policies, i.e. query-based policies, schedule system checks and synchronizations when information queries arrive, which is a stochastics, thus could not avoid interruptions to business operations. (iii) The applied policy is dynamic in nature, which although schedules system checks periodically, incurs system synchronizations only if the system state exceeds the optimal threshold according to the policy. In contrast, existing time-based policies always schedule synchronizations following a predetermined fixed time interval. When unexpected

information changes happen or when there are a small amount of information changes happening, the static time-based policies would incur unnecessary systems synchronization operations leading to unnecessary costs.

In summary, this study applies a time-based dynamic synchronization policy under an infinite planning horizon, which has never been discussed in the literature. The policy adopted can keep the dynamic nature of system maintenance while avoiding any unnecessary operation or disruptions to regular business operations.

Problem Description

To support organizations' strategic shift to analytics, we define the consolidated information gleaned from various sources as the *consolidated data repository.* The consolidated data repository is deployed to respond to information queries from different users, ranging from data scientist, sales representatives, middle to high level managers, and event Business Intelligent applications.

Stochastic Processes of Information Queries and Data Changes

Usually, these information queries arrive randomly to the maintained consolidated data repository. Following the literature (Dey et al., 2006, Qu and Jiang, forthcoming), we assume the arrival of information queries follows a homogeneous Poisson process with arrival rate $\lambda_{\mathrm{Q,h}}$, and the amount of accumulated information queries during time interval (t, t+1) follows the following distribution:

$$\mathrm{Q}_{(t,t+1)}^{h} \sim Poisson\left(\lambda_{\mathrm{Q,h}}\right), h = 1, 2, 3, \dots\dots, H \tag{4-1}$$

In this study, we consider a general application scenario, where there are multiple types (H) of information queries to the organization's consolidated data repository, and each

type of information query follow an independent Poisson distribution. The arrival of information queries and data changes are depicted in Figure 4.1.

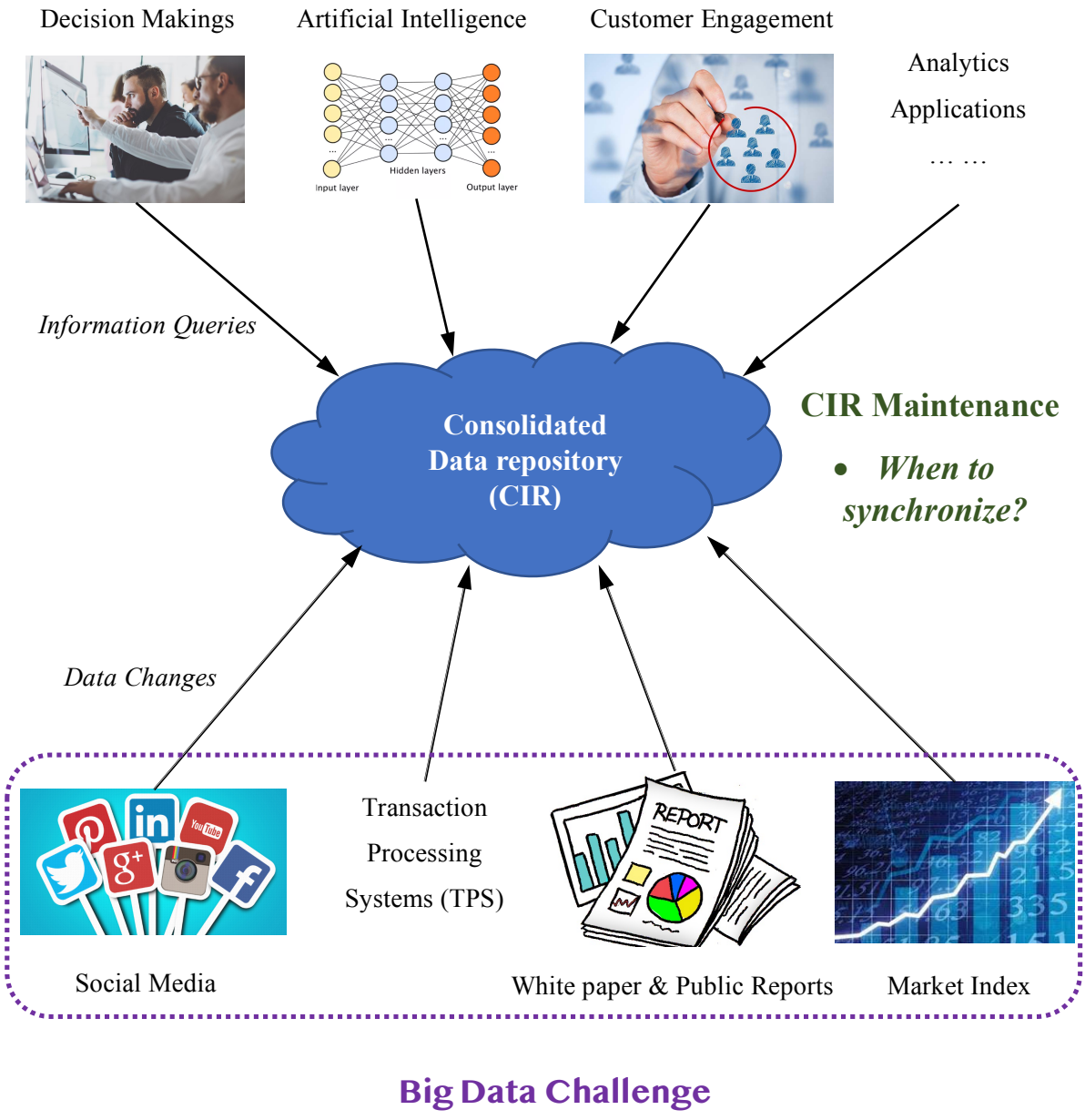


Figure 4.1 Problem Description: Consolidate Data Repository Maintenance

On the other hand, raw data from different sources are collected by organizations and useful information is extracted to help support business requirements from various

information requests. However, facing the challenge of big data, data from these sources keeps changing at real-time. For instance, potential customers post thousands or millions of messages on social media and new business transactions happen in the organizations business process. Walmart, one of the major retailor in the market, processes over 40 Petabytes of data, per day[3]. Meanwhile, information that has been collected and stored in the consolidated data repository may change when records or profiles from the data sources get updated. Customers may return products they have purchased online, posted messages can be removed, and average product price may fluctuate from competitor. If these data changes cannot be incorporated into the consolidated information system in a timely manner, the stored stale information would lead to losses in decision makings. To model the arrival of data changes in the data sources, we also assume each data change follow a homogeneous Poisson process, with arrival rate in each unit time interval $\lambda_{\Gamma,g}$. Therefore, the accumulated type g data changes during time interval (t, t+1) follows:

$$\Gamma_{(t,t+1)}^{g} \sim Poisson(\lambda_{\Gamma,g}), g = 1, 2, 3, \dots\dots, G \quad (4\text{-}2)$$

System States and Actions

Given the stochastic arriving processes of data changes and information queries, the maintenance process can be modeled as a Markov decision process. When an information query arrives, what really impacts the quality of the responding information is the condition of the information stored in the consolidated data repository. In other words, it is the accumulated number of data changes or stale information in the system that directly impact the quality of responded information. When did each data change happen, when was the previous synchronization operation, and how many information queries arrived in the history,

[3] https://datafloq.com/read/big-data-walmart-big-numbers-40-petabytes/1175

do not influence when to schedule the next synchronization as long as the accumulated data changes are given. Therefore, we define the system state as the amount of accumulated data changes or stale information:

Definition 1. System State

The state of a Consolidated Data repository (CIR) is a vector composed of the numbers of accumulated data changes of all types, i.e., $S = (\Gamma^1, \Gamma^2, \dots, \Gamma^g, \dots \Gamma^G)$, $\Gamma^g \in N$, *and the system state space is* $\mathbb{S} = \mathbb{N}^G$.

Based on the definition, the change in system states from time k to k+1, which a difference of one unit time interval, can be presented as $S_{(k,k+1)} = (\Gamma^1_{(k,k+1)}, \Gamma^2_{(k,k+1)}, \dots, \Gamma^g_{(k,k+1)}, \dots, \Gamma^G_{(k,k+1)})$. Given the system state, since data changes follow homogeneous Poisson processes, the system state transition probability can be derived. Each type data change $\Gamma^g_{(k,k+1)}$ follows a Poisson process with an arrival rate, $\lambda_{\Gamma,g}$, and each data change arrive independently. As a result, the joint probability distribution of the incremental system state change $S_{(k,k+1)}$ can be denoted as:

$$p\left(S_{(k,k+1)}\right) = \frac{(\lambda_{\Gamma,1}I)^{\Gamma^1_{(k,k+1)}} e^{-\lambda_{\Gamma,1}I}}{\Gamma^1_{(k,k+1)}!} * \frac{(\lambda_{\Gamma,2}I)^{\Gamma^2_{(k,k+1)}} e^{-\lambda_{\Gamma,2}I}}{\Gamma^2_{(k,k+1)}!} * \dots * \frac{(\lambda_{\Gamma,G}I)^{\Gamma^G_{(k,k+1)}} e^{-\lambda_{\Gamma,G}I}}{\Gamma^G_{(k,k+1)}!} \quad (4\text{-}3)$$

Following the time-based dynamic synchronization policy, the consolidated data repository is checked periodically, but synchronization operation is only run when it is optimal. Therefore, at each decision epoch (or each check point), the action space consists of two optional actions, to synchronize or not to synchronize.

Definition 3. Action Space

The action space at each system check point is $A = \{0, 1\}$, *where* $a_t = 1$ *means to synchronize and* $a_t = 0$ *means not to schedule the synchronization operation*. *Further*, $a_t < a_t'$ *if and only if* $a_t = 0$ *and* $a_t' = 1$.

The optimal system maintenance policy should be developed to choose the optimal action, i.e. synchronize or not, given the system state at each decision epoch.

Economic Cost Analysis

In the maintenance of the consolidated data repository, there are mainly two types of economic costs related, specifically, the *synchronization cost* and the *information staleness cost*.

On the one hand, running system synchronization is not free. Organizations need to spend a fixed amount of investment on hardware and software for the system maintenance. When synchronization is scheduled, labor work is needed to monitor the synchronization operation. In addition, when the consolidated data repository is under synchronization, the indices and materialized views are usually taken offline for updates. When an information query arrives, it cannot get immediate response from the system. Therefore, decisions to be made based on this query get delayed. Such an opportunity cost should also be considered. Beyond the fixed cost for system synchronization, the time needed for synchronization also depends on the size of unprocessed data changes. Usually, larger data changes take longer. However, no matter what policy is adopted in scheduling the system maintenance, the data changes and the variant processing cost should be the same across different schedules and do not impact the optimal schedules. As a result, this proportion of variable cost will not be modeled in the problem. A constant synchronization cost, C_S, is assumed in the model.

On the other hand, when the consolidated information system is not maintained properly, information stored in the system will be largely outdated. When information queries are fed with stale information, it will lead to low-quality decision-makings. Sales prediction becomes in accurate and potential customers cannot be precisely identified. Strategical and operational decisions cannot be make effectively, which eventually will make the business suffer. Therefore, we define information staleness cost as business losses or opportunity costs incurred by stale information whenever a query arrives. All data changes happening before the arrival of a query will lead to losses related to that information query. Based on this setup, we define stale information cost as follows:

Definition 2. Information Staleness Cost

Different types of data changes lead to stale information stored in the consolidated data repository, which causes staleness costs to different information queries independently, and the unit information staleness cost caused by one occurrence of one specific type of data change to one specific type of information query is fixed.

Based on the definition, unit information staleness cost is incurred independently and constant over time, we denote one unit type g data change will incur $\beta_{g,h}$ cost to one type h information query. As a result, when the consolidated data repository is in state $S = (\Gamma^1, \Gamma^2, \dots, \Gamma^g, \dots \Gamma^G)$, when a type h information query arrives, the staleness cost that will be incurred will be $f_h(S) = \sum_{g=1}^{G} \beta_{g\to h} \Gamma^g$.

The system state define earlier is a multi-dimension notation for the information staleness in the system, which is not directly comparable given two different system states. Following the definition about the information staleness cost, we can measure the status of the consolidated data repository from the economic perspective by how much business losses

it may incur. Based on the time-based dynamic synchronization policy, the system will be checked every *I* time interval. We can evaluate the system status at the beginning of a time interval based on the cost that may be incurred during the next check interval. Assume n_h is the number of type *h* query during the check interval, where *h*= 1, 2, 3, …, *H*. So the cost incurred by system state *S* during the next interval would be:

$$C(S) \equiv C_{(t,t+I)}(S) = \sum_{h=1}^{H} n_h * f_h(\Gamma^1, \Gamma^2, \dots, \Gamma^g, \dots \Gamma^G) = \sum_{h=1}^{H} [n_h * \sum_{g=1}^{G} f_{g \to h}(\Gamma^g)\ ] \quad (4\text{-}4)$$

$C_{(t,t+I)}(S)$ is the *current information staleness cost* incurred by system state *S*. Since the amount of information queries follow a Poisson process during one check interval, $n_h \sim Poisson(\lambda_{Q,h} * I)$, we can derive the *expected current information staleness cost* as:

$$E[C(S)] = \sum_{h=1}^{H} \left[\lambda_{Q,h} * I * \sum_{g=1}^{G} f_{g \to h}(\Gamma^g)\ \right] = \sum_{g=1}^{G} \Gamma^g (\sum_{h=1}^{H} \lambda_{Q,h} I \beta_{1 \to h}) \quad (4\text{-}5)$$

Therefore, by the expected current information staleness cost, we can reduce the *H* dimension system state vector into a scaler for further comparison.

Definition 2. System State Order

The order of system state $S^1 < S^2$ *if and only if* $E[C(S^1)] < E[C(S^2)]$, *and* $S^1 = S^2$ *if and only if* $E[C(S^1)] = E[C(S^2)]$, *where* $S^1, S^2 \in \mathbb{S}$ *and* $\mathbb{S} = N^G$.

In other words, the repository's system state is defined by the severity of the information staleness problem, measured by the expected economic costs that may be incurred.

There is a third type of cost to consider in the consolidated data repository maintenance, the *business disruption cost*. When system synchronization is scheduled during business hours, the synchronization operation would compete for computing resources (e.g. hardware, software, and human labor), which may slow down the system of regular business

operations. For instance, A slowdown of a page load by just one second would cost Amazon $1.6 billion in sales and a delay of 0.4 second in displaying the search results could cost Google.com 8 million searchers one day (Eaton, 2012). Therefore, the disruption cost needs to be incorporated during the policy development (Dey et al., 2015). However, in the time-based dynamic synchronization policy, system checks can be scheduled off business hours. It will not influence the optimal synchronization schedule in this study.

In summary, based on the assumptions and definitions discussed above, we derive the optimal maintenance policy for the infinite planning horizon in the next section.

Optimal Maintenance Policy under an Infinite Horizon

In general, the consolidated data repository is considered a long-term organizational IT infrastructure for the strategic shift to analytics, and decision-makers may not foresee an endpoint when deciding the maintenance policy. In this case, it is reasonable to consider an infinite time horizon. We next develop the optimal maintenance policy under an infinite planning horizon.

A Markov Decision Process Model

The Poisson arrival assumption enables us to formulate the database update problem as a *Markov decision process* (Bellman 1957) since the system state is defined as the accumulated amount of various types of data changes. The historical changes and queries arrivals do not directly influence further synchronizations.

At each system check point k, the system status is monitored and the policy should decide whether it is necessary to run the synchronization.

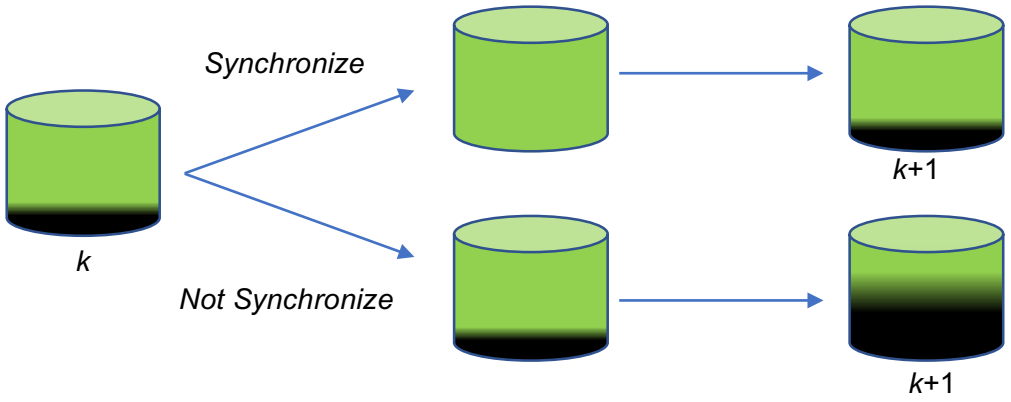


Figure 4.2 A Markov Decision Process of CDR Maintenance

When the decision is to synchronize, we assume the data repository will become error-free immediately, although there will a synchronization cost, C_S, incurred. Even the data repository is cleaned right after decision epoch k, till the next decision epoch k+1, there will be data changes happening during the time interval, leading to information staleness costs for information queries after the cumulated changes. This proportion cost is defined as interval information staleness cost. Based on the literature (Qu and Jiang, forthcoming), the expected interval information staleness cost can be derived as: $E\left[C_{(k,k+1)}\right] = \frac{I^2}{2}\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)$.

If the decision is not to synchronize, the accumulated stale information will remain in the consolidated data repository and cause business losses to future arriving information queries. This cost can be measured as: $E[C(S_k)] = \sum_{g=1}^{G}\Gamma_k^g\left(\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}I\beta_{g\to h}\right)$, the expected current information staleness cost, which only depends on the repository's status and the check interval. At the same time, during the next check interval, the expected interval information staleness cost, $E\left[C_{(k,k+1)}\right]$, will be incurred.

The discussions above is the cost incurred during each check interval. To calculate the total maintenance cost (including both synchronization cost and information staleness cost), different criteria, such as expected total reward, expected average reward, and expected total discounted reward, can be used to evaluate dynamic policies under an infinite time horizon. Here, the expected total reward is not an option because the result will be infinite under an infinite time horizon. For long-term business decisions, the discount factor is commonly considered. Therefore, we decide to adopt the *expected total discounted system cost* criterion to compare the performance of different update policies and select the best policy.

For an infinite Markov decision process, at each decision epoch, we need to assess the present interval cost, i.e., for the interval immediately following a decision epoch, associated with a decision as well as its impact on all future costs for the infinite time horizon.

The present interval cost, denoted by C_{pi}, consists of the current state staleness cost or the update cost (depending on the action at decision epoch), and the interval staleness cost. Hence,

$$C_{pi}(k, S_k, a_k) = a_k C_U + (1 - a_k)E[C(S_k)] + E\left[C_{(k,k+1)}\right] \tag{4-6}$$

To calculate the expected total system maintenance cost, a unit-time discount factor θ is introduced to discout future costs into current value. Therefore, following any synchronization policy π, the expected total discounted system cost for an infinite decision horizon is:

$$V_\pi(S_k) = E\left[\sum_{k=0}^{\infty} \theta^{kI} C_{pi}(k, S_k, a_k) \,\middle|\, \pi, S_k\right] \tag{4-7}$$

The equation above can also be derived into the form of a Bellman equation:

$$V_\pi(k,S_k) = \min_{a_k \in A_s} \{C_{pi}(k,S_k,a_k) + \sum_{S_{k+1}\in \mathbb{S}} \theta^I p(S_{k+1}|S_k,a_k) V_\pi(k+1,S_{k+1})\} \qquad (4\text{-}8)$$

$V_\pi(k,S_k)$ is the total expected system cost starting from decision epoch k given the repository in state S_k. The future cost is the total expected system cost from the next decision epoch on, i.e., $E[V_{k+1}(S_{(k,k+1)} + (1-a_k)S_k)]$, which implicitly includes all future (discounted) costs. $p(S_{k+1}|S_k,a_k)$ is the system transition probability from state $S_k = \left(\Gamma_k^1, \Gamma_k^2, \dots, \Gamma_k^g, \dots, \Gamma_k^G\right)$ to state $S_{k+1} = (\Gamma_{k+1}^1, \Gamma_{k+1}^2, \dots, \Gamma_{k+1}^g, \dots, \Gamma_{k+1}^G)$ given action a_k.

When the repository is synchronized at epoch k, $a_k = 1$, the repository will become error-free immediately, and the probability of transiting to state S_{k+1} is the probability that data changes during the next check interval accumulated to $(\Gamma_{k+1}^1, \Gamma_{k+1}^2, \dots, \Gamma_{k+1}^g, \dots, \Gamma_{k+1}^G)$, which is $\frac{(\lambda_{\Gamma,1}I)^{\Gamma_{k+1}^1} * \dots * (\lambda_{\Gamma,G}I)^{\Gamma_{k+1}^G} * e^{-\sum_{g=1}^G \lambda_{\Gamma,g}I}}{\Gamma_{k+1}^1! * \Gamma_{k+1}^2! * \dots * \Gamma_{k+1}^G!}$.

When the consolidated data repository is not synchronized at epoch k, $a_k = 1$, the repository will accumulate from state S_k to state S_{k+1}. The probability for the transition is each type data change g accumulate $(\Gamma_{k+1}^g - \Gamma_k^g)$ changes during the next check interval. Since each data change type arrives independently, the probability is $\frac{(\lambda_{\Gamma,1}I)^{(\Gamma_{k+1}^1 - \Gamma_k^1)} * \dots * (\lambda_{\Gamma,G}I)^{(\Gamma_{k+1}^G - \Gamma_k^G)} * e^{-\sum_{g=1}^G \lambda_{\Gamma,g}I}}{(\Gamma_{k+1}^1 - \Gamma_k^1)! * \dots * (\Gamma_{k+1}^G - \Gamma_k^G)!}$.

So in general, the state transition probability can be defined as follows:

$$p(S_{k+1}|S_k,a_k) = \begin{cases} \frac{(\lambda_{\Gamma,1}I)^{\Gamma_{k+1}^1} * \dots * (\lambda_{\Gamma,G}I)^{\Gamma_{k+1}^G} * e^{-\sum_{g=1}^G \lambda_{\Gamma,g}I}}{\Gamma_{k+1}^1! * \Gamma_{k+1}^2! * \dots * \Gamma_{k+1}^G!}, & if\ a_k = 1 \\ \frac{(\lambda_{\Gamma,1}I)^{(\Gamma_{k+1}^1 - \Gamma_k^1)} * \dots * (\lambda_{\Gamma,G}I)^{(\Gamma_{k+1}^G - \Gamma_k^G)} * e^{-\sum_{g=1}^G \lambda_{\Gamma,g}I}}{(\Gamma_{k+1}^1 - \Gamma_k^1)! * \dots * (\Gamma_{k+1}^G - \Gamma_k^G)!}, & if\ a_k = 0 \end{cases} \qquad (4\text{-}9)$$

Another constraint for the above transition probability is Γ_{k+1}^{g} cannot be smaller than Γ_{k}^{g} when there is no synchronization, $\Gamma_{k+1}^{g} \geq \Gamma_{k}^{g}\ if\ a_k = 0$. In other words, the data repository cannot become reduce the level of staleness in each type of data change unless a synchronization.

Existence of an Optimal Stationary Synchronization Policy

Under an infinite planning horizon, the decision at any decision epoch *k* is *Markov-Deterministic* (MD) — given the system state S_k, the action taken is deterministic: $a_k = d(S_k) = \pi^{MD}(S_k)$. Furthermore, the optimal system cost function is bounded because it can be shown that

$$V_\pi(S_k) \leq E\left[\sum_{t=0}^{\infty} \theta^{tI} {C_{pi}}^{max} \middle| \pi^{MD}, S_k\right] = \frac{1}{1-\theta^I} {C_{pi}}^{max},$$

$$\text{where } {C_{pi}}^{max} = C_U + \frac{I^2}{2}\left[\sum_{h=1}^{H} \lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G} \beta_{g\to h}\lambda_{\Gamma,g}\right)\right] \qquad (4\text{-}10)$$

More importantly, given the homogeneous arrival processes of data changes and information queries, we can show that the optimal CDB update policy is stationary.

Lemma 1. *The optimal CDB update policy in an infinite time horizon, if exists, is stationary.*

In fact, in an infinite time horizon, the expected total discounted system cost at a decision epoch does not depend on time, hence the optimal system cost function for a stationary policy $\pi = (d, d \ldots)$ is:

$$V_\pi(S) = C_{pi}\big(S, d(S)\big) + \theta^I \sum_{S' \in \mathbb{S}} T_p[S, S']\, V_\pi(S') \qquad (4\text{-}11)$$

where T_p is the probability transition matrix, and each element in the matrix, $T_p\ [S, S'] = p\ (S'|S, d(S))$, represents the probability of transition from state S to state S' under a given policy π.

According to Puterman (2005, p. 153), when the state space is discrete and the supremum or infimum is attainable, there exists a *unique optimal deterministic stationary policy*. Since such conditions are satisfied in the consolidated data repository maintenance problem, we have the following conclusion:

Proposition 1. *There exists a unique optimal deterministic and stationary consolidated data repository synchronization policy in an infinite time horizon.*

In sum, in the infinite-time horizon, the optimal synchronization policy exists and is stationary. The decision rules at different decision epochs are the same, i.e., $\pi^* = (d^*, d^*, \dots) = d^{*\infty}$.

Expected Total Discounted System Maintenance Cost

To evaluate a synchronization policy in an infinite time horizon, we need to compute the expected total discounted system cost under the policy for all possible system states. With all possible system states considered, Eq. (4-11) can be taken as a linear system with variables $V_\pi(S^1), V_\pi(S^2), \dots, V_\pi(S^{ub})$, and each equation for a given system state is a constraint.

Variables in the linear system can be vectorized as:

$$\overrightarrow{V_\pi} = [V_\pi(S^1), V_\pi(S^2), \dots, V_\pi(S^{ub})] \tag{4-12}$$

$$\overrightarrow{C_{pi}} = \left[C_{pi}\left(S^1, d(S^1)\right), C_{pi}\left(S^2, d(S^2)\right), \dots, C_{pi}\left(S^{ub}, d(S^{ub})\right)\right] \tag{4-13}$$

Therefore, based on Eq. (4-11), we can use linear algebra to solve the optimal value function:

$$\overrightarrow{V_\pi} = \overrightarrow{C_{pi}} + \theta^I T_p \overrightarrow{V_\pi} \tag{4-14}$$

Here T_p is a transition matrix: $T_p\ [S, S'] = p\ (S'|S, d(S))$, and each element in the matrix is defined in Eq. (4-8) . From (4-14), we have

$$\overrightarrow{V_\pi} = \left(I_{|\mathbb{S}|} - \theta^I T_p\right)^{-1} \overrightarrow{C_{pi}} \tag{4-15}$$

Because T_p is the state transition probability matrix, in which each element value is no larger than 1, it can be taken as a linear transformation on the normed linear space. The discount factor $\theta \in (0,1)$. Hence the values in matrix $\theta^I T_p$ should be strictly less than 1, i.e. $\sigma\left(\theta^I T_p\right) < 1$. Thus matrix $\theta^I T_p$ is a bounded linear transformation on a Banach space. Since $\theta^I T_p$ is a bounded linear transformation on a Banach space and $\sigma\left(\theta^I T_p\right) < 1$, the inversion of $\left(I_{|\mathbb{S}|} - \theta^I T_p\right)$ exists.

Given an update policy $\pi = d^\infty$, the present interval cost vector $C_{pi}\left(S, d(S)\right)$ and the transition probability matrix T_p can be constructed. Subsequently we can obtain the optimal system cost vector based on Eq. (4-15). Because it contains the expected total discounted system cost, the optimal expected system cost vector $\overrightarrow{V_\pi}$ can help evaluate all feasible update policies.

Search for the Optimal Stationary Update Policy

The system cost computation presented in the previous subsection assumes that an update policy π is given. For such a policy to be optimal, at each decision epoch, its decision rule should determine whether to update the CDB or not based on which action leads to a lower cost for a given system state:

With an update, the expected total discounted system cost is:

$$V(S, a = 1) = C_U + \theta^I \sum_{S' \in \mathbb{S}} p(S' \mid S, a = 1)\, V\ (S'\ ) + E\left[C_{(0,I)}\right] \tag{4-16}$$

In the absence of an update, the cost is:

$$V(S, a=0) = E[C(S)] + \theta^I \sum_{S' \in \mathbb{S}} p(S' \mid S, a=0)\, V(S') + E\left[C_{(0,I)}\right] \quad (4\text{-}17)$$

The optimal stationary policy $\pi^*(S)$ should always select the action that leads to a smaller cost. That is, the optimal policy $\pi^* = d^{*\infty}$ should take the form,

$$d^*(S) = \begin{cases} 1 & if\ V(S, a=0) - V(S, a=1) \geq 0 \\ 0 & if\ V(S, a=0) - V(S, a=1) < 0 \end{cases} \quad (4\text{-}18)$$

Lemma 2. *In an infinite time horizon, the expected total discounted system cost $V(S)$ is a non-decreasing function of the system state S, i.e., if $S \geq S'$, $V(S) \geq V(S')$.*

Lemma 2 shows that, similar to the finite-horizon scenario, in an infinite time horizon, a consolidated data repository with more severe information staleness problems typically leads to a higher expected system cost.

Lemma 3. *The action recommended by the optimal time-based dynamic synchronization policy ($\pi^* = d^{*\infty}$) for an infinite time horizon is non-decreasing with S, i.e., $d^*(S) \geq d^*(S')\ if\ S \geq S'$.*

Without synchronization operation, the consolidated data repository will deteriorate in terms of the cumulated number of data errors. Similar to the case under a finite time horizon, when the state of the data repository reaches a threshold, it becomes necessary to run the synchronization.

Proposition 2. *Under an infinite time horizon, there exists a threshold η^*, such that at any decision epoch, it is optimal not to update the consolidated data repository if $E[C(S)] < \eta^*$, and update the CDB otherwise.*

Proposition 2 implies an optimal decision rule in the form of

$$a(S) = d^*(S) = \begin{cases} 1, & E[C(S)] \geq \eta^* \\ 0, & E[C(S)] < \eta^* \end{cases} \quad (4\text{-}19)$$

On the surface, the only difference between the optimal decision rules for the finite horizon in the literature and rule (4-19) is that the latter is stationary or time-independent. The algorithms used to find the optimal threshold, however, are very different under the two cases.

The threshold of the optimal policy should be generated by comparing $V(S, a = 1)$ against $V(S, a = 0)$. However, before knowing the synchronization policy (essentially the threshold for $E[C(S)]$), the value function $V(S)$ cannot be computed. Therefore, we develop an efficient algorithm, as summarized in Figure 4.3, to search for the optimal threshold η^*.

Briefly, the algorithm starts by assigning a threshold value η; given this threshold value, it can decide whether to synchronize a consolidated data repository or not by comparing $E[C(S)]$ with η according to rule (24). With the synchronization decision known, the present interval cost $C_{pi}(S)$ can be calculated. Repeating this step for every possible system state *S* generates the present interval cost vector $\overrightarrow{C_{pi}}$, which, together with the transition probabilities, makes it possible to compute the expected total system cost vector $\overrightarrow{V_{\pi}}$. Given this vector, the algorithm can calculate and compare $V(S, a = 1)$ and $V(S, a = 0)$ to decide whether the consolidated data repository should be synchronized or not. If the synchronization decision reached by this cost comparison method is the same as that reached by comparing $E[C(S)]$ and η, η is the optimal threshold. Otherwise, the value of η is adjusted and the aforementioned calculation and comparison are repeated until the optimal η^* is found.

Input: C_U, I, $\lambda_{\mathrm{Q,h}}$, $\lambda_{\Gamma,\mathrm{g}}$, θ

Output: the optimal threshold η^*

Step 0: a=0, b=C_U.

Step 1: set $\eta = \frac{a+b}{2}$.

Step 2: use η as the threshold for the value of $E[C(S)]$, the policy π is:

-- if $E[C(S)] \geq \eta$, then update, $d(S) = 1$;

-- if $E[C(S)] < \eta$, then no-update, $d(S) = 0$.

Step 3: Follow the policy in step 2, construct the present interval cost vector and the transition matrix *T*, get the optimal system cost vector $\overrightarrow{V_\pi}$.

Step 4: For all possible states *S* in $\mathbb{S}$, if $E[C(S)] - \eta$ has the same sign with $E[C(S)] + E\left[V_\pi\left(S + S_{(k,k+1)}\right)\right] - E\left[V_\pi\left(S_{(k,k+1)}\right)\right] - C_U$, then the optimal policy is π^* with threshold η, break.

Otherwise:

(1) If $E[C(S)] - \eta > 0$, while $E[C(S)] + E\left[V_\pi\left(S + S_{(k,k+1)}\right)\right] - E\left[V_\pi\left(S_{(k,k+1)}\right)\right] - C_U < 0$, the threshold η is too low, let $a = \eta$, $b = b$, go to step 1;

(2) If $E[C(S)] - \eta < 0$, while $E[C(S)] + E\left[V_\pi\left(S + S_{(k,k+1)}\right)\right] - E\left[V_\pi\left(S_{(k,k+1)}\right)\right] - C_U > 0$, the threshold η is too high, let $a = a$, $b = \eta$, go to step 1.

Figure 4.3 Threshold Searching Algorithm Under Infinite Horizon

Once the optimal threshold η^*, and hence the optimal policy π^*, is found, the optimal action at each epoch can be determined by applying decision rule (4-19). This time-based threshold policy for an infinite time horizon is easy to operationalize because it checks the system state based on a predetermined time schedule, the expected current state staleness cost is easy to calculate, and the same threshold is used for all decision epochs.

Policy Comparisons

In this section, we will evaluate the performance of the applied time-based dynamic synchronization policy under an infinite time horizon by the total maintenance incurred by different maintenance policies.

Optimal Check Interval

Based on the time-based dynamic synchronization policy, the data repository will be checked following a predetermined time interval. In the literature (Qu and Jiang, forthcoming), the boundary of the feasible check intervals has been developed. The minimal check interval is usually determined by operation and technical constraints, while the upper bound of the check interval can be developed analytically as $I^{ub} = \sqrt{C_U\left[\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)\right]^{-1}}$. They find that the minimal check interval is always the optimal.

Although checking the status of the consolidated data repository as frequently as possible can help track every small data change close to real time, system status checks are not cost-free. In practice, identifying data changes and extracting those changes are time-consuming tasks in an ETL process.

For instance, checking the system status of a consolidated information system may require checking data changes in source systems before the ETL process. For most source systems, identifying the recently modified records is difficult or intrusive to the operation of the system (Parida, R. 2008). Typically, to efficiently keep track of data changes, it is necessary to implement a customized tracking method on source systems, which uses a combination of triggers, timestamp columns, and additional tables to identify data changes.

Creating such applications incurs development cost, and running them leads to computational overhead.

Each time a check is performed, certain cost, such as the overhead of switching from one process to another in a system and loading a tracking method, will be incurred regardless of the amount of data changes, while other cost could be dependent on the amount of data changes. Therefore, we assume *system check cost* consists of a fixed check cost and a variable check cost that is a linear function of the size of data changes. Regardless of the adopted policies, the total accumulated data changes to be collected during the planning horizon, and hence the total variable check cost, are the same. Therefore, only fixed check cost need to be considered for policy evaluation and check interval selection. We denote the fixed check cost by C_{check} for the time-based dynamic synchronization policy as well as benchmark policies.

The shorter the check interval, the more checkpoints (decision epochs) will result. Intuitively, more system checks can improve the maintenance and reduce the sum of information staleness and synchronization costs. On the other hand, frequent system checks lead to higher check cost. Consequently, an optimal check interval should be determined by minimizing the sum of all costs.

For the time-based dynamic synchronization policy, the system check cost is always incurred, hence it does not affect the optimal action at each decision epoch, but it impacts the selection of the optimal system check interval. To understand its impact, we try different check costs, and plot how the total system cost changes with the length of check interval, as shown in Figure 4-4.

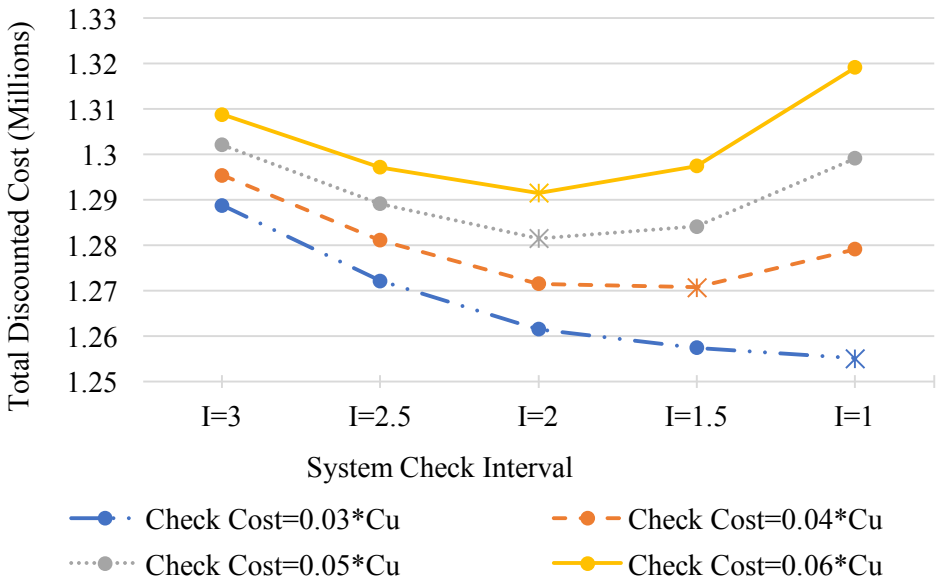


Figure 4.4 Maintenance Cost with Different System Check Cost

The non-monotonic curves in Figure 4.4 show that the minimum check interval is not automatically the optimal check interval when system check cost is considered. To decide the optimal check interval, we need to numerically calculate the expected total system costs associated with different check intervals in the feasible range [*minimal check interval*, *maximal check interval* (I^{ub})]. Here we would like to emphasize that it is generally not necessary to check a large number of feasible intervals. For most practical applications, we believe that feasible check intervals should consist of multiples of days or hours. Additionally, the selection of system check interval should also avoid rush business hours. Once the feasible check intervals are decided, numerical analysis can be conducted to select the optimal check interval.

Policy Comparison under Infinite Horizon

Regarding the time-based dynamic synchronization policy, the optimal threshold can be obtained using the search algorithm summarized in Figure 4.3.

Because both the periodic policy and time-based dynamic synchronization policy are stationary in nature in an infinite time horizon, we can compare the performances of the two policies in a truncated finite time (i.e. one year to ten years). We use two type of data errors and two types of queries for illustration. Based on the assumptions, the costs incurred to different queries by the different data errors are independent, hence the comparison involving multiple types of data changes and queries should follow similar patterns.

The key parameter values are set as $\theta = 0.999$, $C_U = 2000$, $\lambda_{Q,1} = 1$, $\lambda_{Q,2} = 3$ $\lambda_{\Gamma,1} = 0.5$, $\lambda_{\Gamma,2} = 1.5$, $\beta_{1\to1} = 120$, $\beta_{1\to2} = 90$, $\beta_{2\to1} = 60$, $\beta_{2\to2} = 30$, I=1, and $C_{check} = 1\%C_U$ in our experiment.

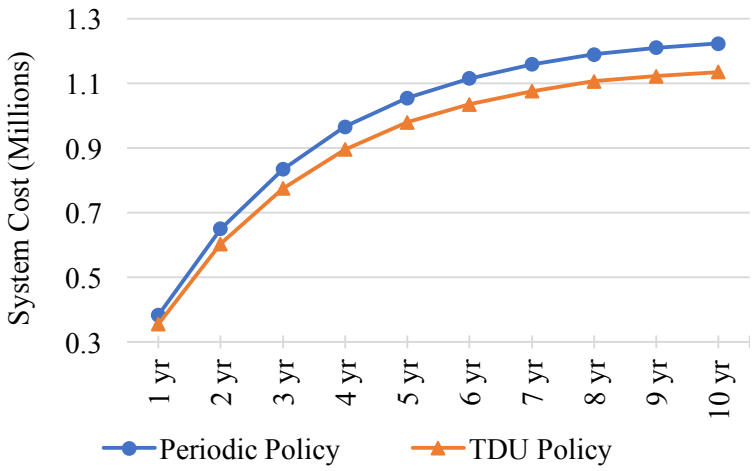


Figure 4.5 Total Maintenance Cost by Periodic Policy and Time-based Dynamic Synchronization Policy Under Infinite Horizon

Figure 4.5 shows that the time-based dynamic synchronization policy dominates the benchmark periodic policy. In Figure 4.5, the proposed dynamic update policy consistently outperforms the benchmark periodic update policy in terms of total expected discounted system cost.

Figure 4.6shows that the percentage of improvement achieved by the TDU policy over the periodic policy remains relatively stable at over 7%.

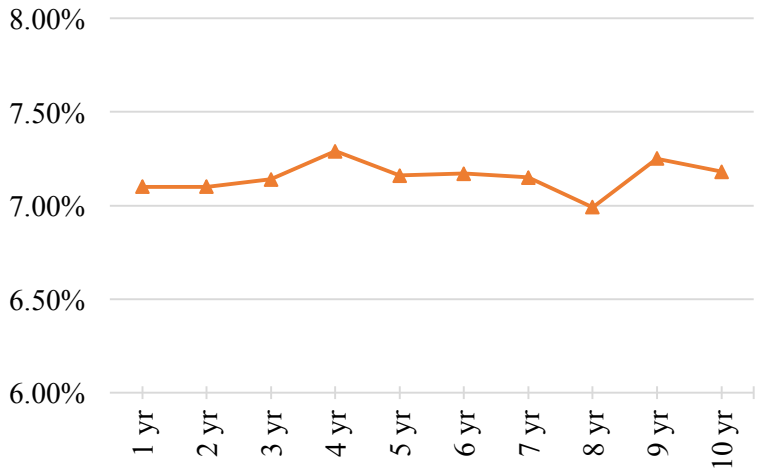


Figure 4.6 Percentage of Cost Savings by the Time-based Dynamic Synchronization Policy

In conclusion, the adopted time-based dynamic synchronization policy performs consistently better than the periodic policy under an infinite time horizon.

Conclusions

Data has become a more important strategic asset for organizations than ever. To support information queries from multiple users within the organization, a consolidated data repository is usually set up to respond to information request regarding every aspect of their business. However, in the age of big data, fast changing business environment brings critical challenges to the maintenance of the consolidated data repository.

The information staleness cost incurred by inaccurate decisions due to outdated information and the synchronization cost are the two major cost factors in the maintenance process. Striking a balance between these two types of economic cost, this study extends the time-based dynamic synchronization to an infinite planning horizon. Different from the finite time horizon, the optimal policy is static over time in an infinite horizon. To achieve the

optimal control policy, a searching algorithm is proposed in this study to efficiently search for the optimal threshold. In addition, the optimal system check interval is further discussed based on the theoretical feasible range. In the policy comparison, we compare the performance of the proposed dynamic maintenance with a static periodic policy. Results show that our optimal control limit policy always outperforms the benchmark method. The cost savings are over 7% even under a very conservative setting, which would translate to a huge among of financial savings in real world.

In summary, the policy and the optimal control searching algorithm is easy to operationalize in real-worlds settings and can significantly reduce the maintenance cost. For further research, some hybrid policies can be developed to retain the advantages of different policies. In addition, the parameters of different types of information staleness cost can be estimated empirically from applications, which could bring more business insights to system mangers.

References

Jarke, M., M. Lenzerini, Y. Vassiliou, P. Vassiliadis. 2000. "Fundamentals of Data Warehouses." Springer, New York.

Bellman, R. E. 1957. "A Markovian Decision Process," No. P-1066, Santa Monica, CA: RAND Corporation.

Dey, D., Lahiri, A., and Zhang, G. 2015. "Optimal Policies for Security Patch Management," *INFORMS Journal on Computing* (27:3), pp. 462-477.

Dey, D., Zhang, Z. and De P. 2006. "Optimal Synchronization Policies for Data Warehouses," *INFORMS Journal on Computing* (18:2), pp. 229-242.

Eaton, K. 2012. "How One Second Could Cost Amazon $1.6 Billion in Sales," Fast Company (https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales; accessed May 17, 2018).

Fang, X., Sheng, O. R. L., and Goes, P. 2013. "When is the Right Time to Refresh Knowledge Discovered from Data?" *Operations Research* (61:1), pp. 32-44.

Puterman, M. L. 2005. Markov Decision Processes: Discrete Stochastic Dynamic Programming, New York: John Wiley & Sons.

Qu, X., and Jiang, Z. "A Time-based Dynamic Synchronization Policy for Consolidated Database Systems," *MIS Quarterly*, forthcoming.

Segev, A., and Fang, W. 1991. "Optimal Update Policies for Distributed Materialized Views," *Management Science* (37:7), pp. 851-870.

Xiong, M. and Ramamritham, K., 2004. "Deriving deadlines and periods for real-time update transactions". *IEEE Transactions on Computers* 53(5), pp.567-583.

Zong, W., Wu, F., Jiang, Z. 2017. "A Markov-based update policy for constantly changing database systems." *IEEE Transactions on Engineering Management*, forthcoming.

CHAPTER 5. GENERAL DISCUSSIONS

My dissertation develops and extends consumer analytics tools based on large-scale transaction records to help companies better understand consumers' decision makings, and subsequently predict and influence their consumption behavior.

Findings in my dissertation show recommendations using transactional data usually generate low precision and recall. However, by incorporating the product category information into the recommendation algorithm, the precision and recall improve significantly. In practice, if the product related information can be effectively utilize, it can help alleviate the data sparsity issue. When making upgrade decisions, consumers' previous adoption and usage experience have significant influence on their upgrade timing. Specifically, potential switching customers who are using the latest available generation are more willing to upgrade and heavy users of the product series tend to upgrade earlier. More interestingly, specialized customers (those focusing on a relatively small number of product functions) demonstrate a higher upgrade probability. These findings can help companies better segment their existing users group, increase the precision in target marketing, and enhance consumer engagement. In addition, the data quality is an important factor in analytics applications. The proposed method can effectively schedule the maintenance of consolidated data repository and lead to significant cost savings.

Based on the three studies in the dissertation, there are a few interesting research directions emerging. First of all, since the data quality has a crucial influence on the quality of generated insights and information staleness cost need to be considered when developing analytics tools, it would be interesting to develop some incremental methods to train the model using incremental data changes and then incorporate results from incremental data in

to the main model. This would approximate the real-time analytics while avoiding unnecessary disruptions to business operations. Secondly, the product category information can be integrated in the recommendation algorithm using a similarity measure. In future studies, the hierarchical Bayesian framework can be applied to model the hierarchical product relationship and using structural econometrics models to identify interesting patterns from frequent shopping baskets. Last but not least, a hybrid recommendation system that treats first-time buyers and existing users differently can be developed in more generalized application context.

APPENDIX. PROOFS FOR CHAPTER 4

Proof of Lemma 1

The expected total discounted cost is:

$$V_\pi(k, S_k) = \min_{a_k \in A_s} \{C_{pi}(k, S_k, a_k) + \sum_{S_{k+1} \in \mathbb{S}} \theta^I p(S_{k+1} | S_k, a_k) V_\pi(k+1, S_{k+1})\} \quad (0\text{-}1)$$

In the expected present interval cost function $C_{pi}(k, S_k, a_k) = a_k C_U + (1 - a_k)E[C(S_k)] + E\,[C_{(k,k+1)}]$, the update/current state cost $a_k C_U + (1 - a_k)E[C(S_k)]$ is determined by the system state S_k and the action taken a_k. The interval staleness cost $E\,[C_{(k,k+1)}]$ depends only on the interval length *I*. In addition, the state transition probability $p(S_{k+1}|S_k, a_k)$ is determined by the system state S_k and the interval length *I*.

For decision epoch t_1 and t_2, $t_1 \neq t_2$ but $S_{t_1} = S_{t_2} = S_0$, we first assume they follow different decision rules, i.e., $d_{t_1} \neq d_{t_2}$, then

$$V_\pi(t_1, S_0) = \min_{a_{t_1} \in A_s} \{C_{pi}\big(t_1, S_0, d_{t_1}(S_0)\big) + \sum_{S' \in \mathbb{S}} \theta^I p\big(S' \big| S_0, d_{t_1}(S_0)\big) V_\pi(t_1 + 1, S')\} \quad (0\text{-}2)$$

$$V_\pi(t_2, S_0) = \min_{a_{t_2} \in A_s} \{C_{pi}\big(t_2, S_0, d_{t_2}(S_0)\big) + \sum_{S'' \in \mathbb{S}} \theta^I p\big(S'' \big| S_0, d_{t_2}(S_0)\big) V_\pi(t_2 + 1, S'')\} \quad (0\text{-}3)$$

Since d_{t_1} is the optimal decision rule at time t_1, then

$$C_{pi}\big(t_1, S_0, d_{t_1}(S_0)\big) + \sum_{S' \in \mathbb{S}} \theta^I p\big(S' \big| S_0, d_{t_1}(S_0)\big) V_\pi(t_1 + 1, S') \leq C_{pi}\big(t_1, S_0, d_{t_2}(S_0)\big) + \sum_{S'' \in \mathbb{S}} \theta^I p\big(S'' \big| S_0, d_{t_2}(S_0)\big) V_\pi(t_1 + 1, S'') \quad (0\text{-}4)$$

Because $C_{pi}(k, S_k, a_k) = a_k C_U + (1 - a_k)E[C(S_k)] + E\,[C_{(k,k+1)}]$,

$$d_{t_1}(S_0) C_U + \big(1 - d_{t_1}(S_0)\big) E[C(S_0)] + \sum_{S' \in \mathbb{S}} \theta^I p\big(S' \big| S_0, d_{t_1}(S_0)\big) V_\pi(t_1 + 1, S') \leq d_{t_2}(S_0) C_U + \big(1 - d_{t_2}(S_0)\big) E[C(S_0)] + \sum_{S'' \in \mathbb{S}} \theta^I p\big(S'' \big| S_0, d_{t_2}(S_0)\big) V_\pi(t_1 + 1, S'') \quad (0\text{-}5)$$

At the same time, since d_{t_2} is the optimal decision rule at time t_2, we have

$$C_{pi}\left(t_2, S_0, d_{t_2}(S_0)\right) + \sum_{S'' \in \mathbb{S}} \theta^I p\left(S'' \middle| S_0, d_{t_2}(S_0)\right) V_\pi(t_2 + 1, S'') \leq C_{pi}\left(t_2, S_0, d_{t_1}(S_0)\right) + \sum_{S' \in \mathbb{S}} \theta^I p\left(S' \middle| S_0, d_{t_1}(S_0)\right) V_\pi(t_2 + 1, S') \quad (0\text{-}6)$$

Similarly, the following inequality should always hold:

$$d_{t_2}(S_0) C_U + \left(1 - d_{t_2}(S_0)\right) E[C(S_0)] + \sum_{S'' \in \mathbb{S}} \theta^I p\left(S'' \middle| S_0, d_{t_2}(S_0)\right) V_\pi(t_2 + 1, S'') \leq d_{t_1}(S_0) C_U + \left(1 - d_{t_1}(S_0)\right) E[C(S_0)] + \sum_{S' \in \mathbb{S}} \theta^I p\left(S' \middle| S_0, d_{t_1}(S_0)\right) V_\pi(t_2 + 1, S') \quad (0\text{-}7)$$

Because as long as the system state is the same, $V_\pi(k, S)$ does not depend on starting time k, which means $V_\pi(t_2 + 1, S') = V_\pi(t_1 + 1, S') = V_\pi(S')$ and $V_\pi(t_1 + 1, S'') = V_\pi(t_2 + 1, S'') = V_\pi(S'')$. Therefore, the (A-4) and (A-6) hold at the same time only if the following equation always holds,

$$d_{t_2}(S_0) C_U + \left(1 - d_{t_2}(S_0)\right) E[C(S_0)] + \sum_{S'' \in \mathbb{S}} \theta^I p\left(S'' \middle| S_0, d_{t_2}(S_0)\right) V_\pi(S'') = d_{t_1}(S_0) C_U + \left(1 - d_{t_1}(S_0)\right) E[C(S_0)] + \sum_{S' \in \mathbb{S}} \theta^I p\left(S' \middle| S_0, d_{t_1}(S_0)\right) V_\pi(S') \quad (0\text{-}8)$$

Because (A-8) contradicts the assumption $d_{t_1}(S_0) \neq d_{t_2}(S_0)$, we conclude that the original assumption $d_{t_1}(S_0) \neq d_{t_2}(S_0)$ is invalid. Therefore, we must have $d_{t_1}(S_0) = d_{t_2}(S_0)$, which means the decision rules does not depend on the time of the decision epoch. In other words, the optimal decision policy should be stationary, and the decision rules should remain the same across time, i.e., $d_1 = d_2 = d_3 = \cdots$.

Proof of Lemma 2

We consider two scenarios:

if $d(S) = d(S') = 1$,

$$V(S) = V(S') = C_U + E[C_{(0,I)}] + \theta^I \sum_{S_1} T[S, S_1] V(S_1) \quad (0\text{-}9)$$

If $d(S) \neq d(S')$, assume π is the optimal update policy for system starting with state S, and π' is the optimal update policy for system starting with state S'.

Based on Proof for Lemma 1, we can always have $V_\pi(S') \leq V_\pi(S)$. Since π' is the for state S', $V_{\pi'}(S') \leq V_\pi(S')$. Then we will have $V_{\pi'}(S') \leq V_\pi(S)$.

Based on the discussion above, when $S \geq S'$ *or* $E[C(S)] \geq E[C(S')]$, we can always have $V(S) \geq V(S')$. Therefore, the optimal total maintenance cost is a non-decreasing function of the system state S.

Proof of Lemma 3

The optimal decision rule can take the following form:

$$d^*(S) = \begin{cases} 1 & if\ V(S, a=0) - V(S, a=1) \geq 0 \\ 0 & if\ V(S, a=0) - V(S, a=1) < 0 \end{cases} \quad (0\text{-}10)$$

Suppose $H(S) = V(S, a=0) - V(S, a=1) = E[C(S)] + \theta^I E[V(S+\Delta)] - \theta^I E[V(\Delta)] - C_U = E[C(S)] + \theta^I \sum_{\Delta \in \mathbb{S}} p(\Delta)\,[V(S+\Delta) - V(\Delta)] - C_U$. Then,

$$d^*(S) = \begin{cases} 1 & if\ H(S) \geq 0 \\ 0 & if\ H(S) < 0 \end{cases} \quad (0\text{-}11)$$

Based on the definition of the order of the state space, the expected current state staleness cost $E[C(S)]$ is monotonically increasing with S. In addition, according to Lemma 6, $V(S+\Delta)$ is non-decreasing with S. Overall, $H(S)$ is monotonically increasing with system state S.

Given $S \geq S'$, we could have $H(S) \geq H(S')$.

- If $H(S) \geq H(S') \geq 0$, then $d^*(S) = d^*(S') = 1$;
- If $H(S) \geq 0 > H(S')$, then $d^*(S) = 1$ with $d^*(S') = 0$;

- If $0 > H(S) \geq H(S')$, then $d^*(S) = d^*(S') = 0$.

In summary, $d^*(S) \geq d^*(S')$ will always hold as long as $S \geq S'$. Therefore, the optimal decision rule is non-decreasing with system state S.